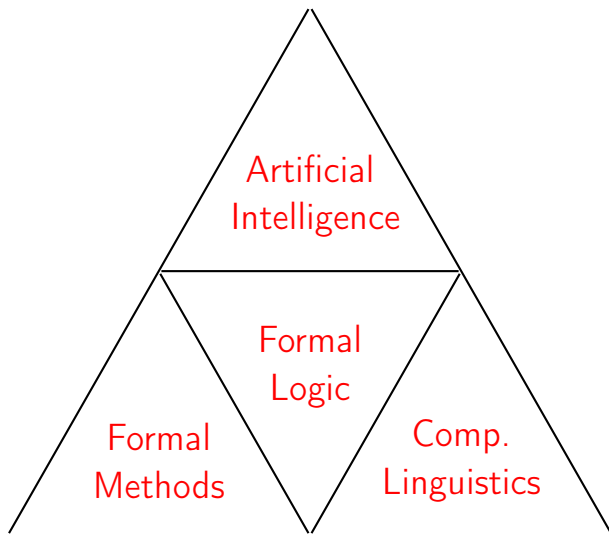


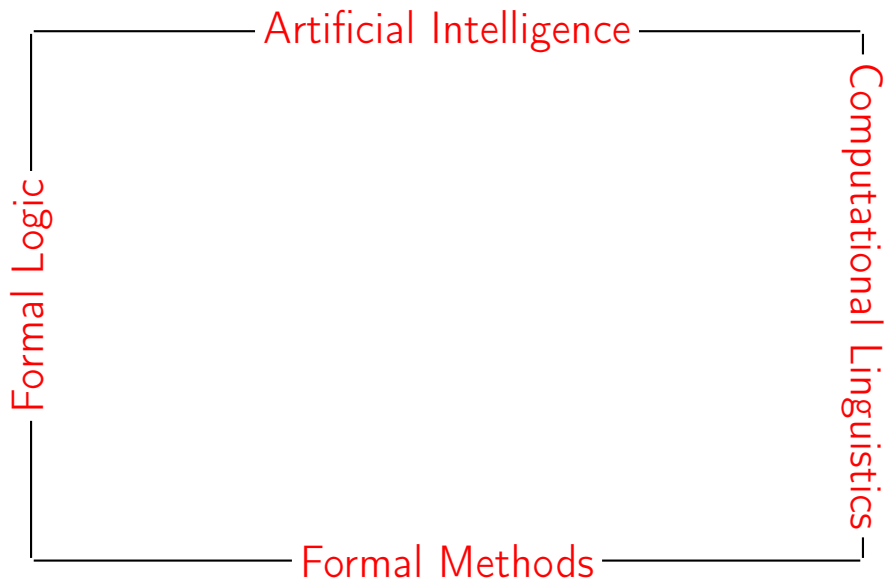
Automating Expressive Non-classical Logics and their Combinations in Classical Higher Order Logic

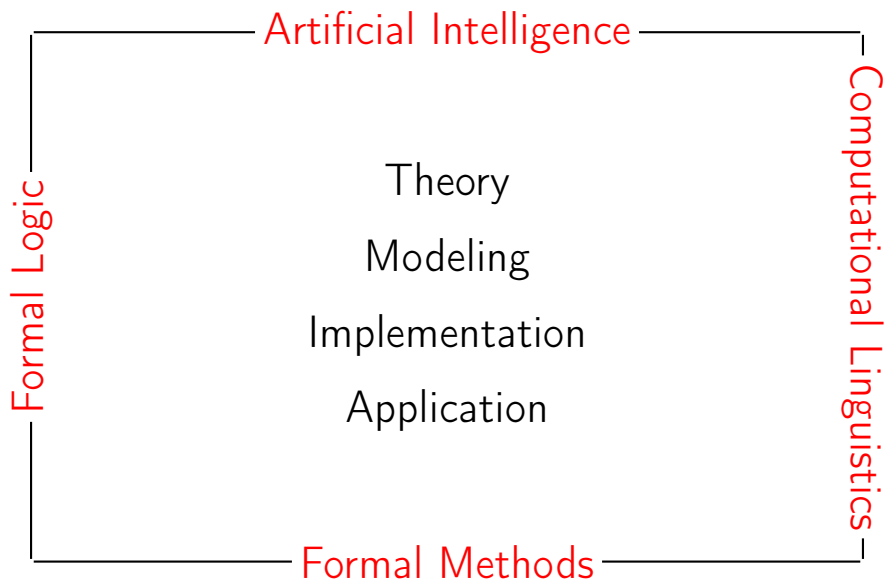
Christoph Benzmüller

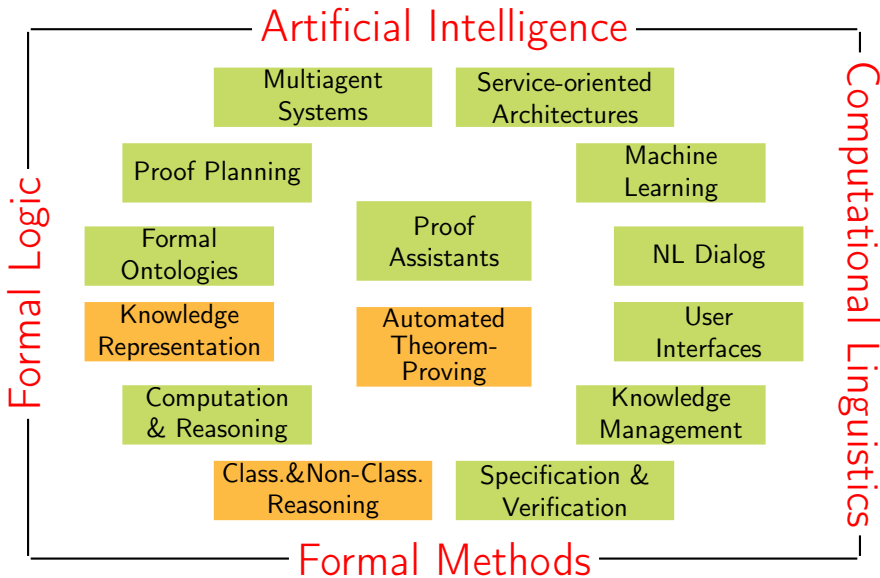
FU Berlin

Presentation at Potsdam University on November 15, 2011









Core Questions:

- 1 Classical Higher Order Logic (HOL) as Universal Logic?
- 2 HOL Provers & Model Finders as Generic Reasoning Tools?
- 3 Combinations with Specialist Reasoners (if available)?

Core Questions:

- 1 Classical Higher Order Logic (HOL) as Universal Logic?
- 2 HOL Provers & Model Finders as Generic Reasoning Tools?
- 3 Combinations with Specialist Reasoners (if available)?

Outline:

- What is HOL?
- Mechanization & Automation of HOL
- Examples of Natural Fragments of HOL: Multimodal Logics & Others
- Automation of Logics and Logic Combinations in HOL
- Automation of Meta-Properties of Logics in HOL
- Conclusion



What is HOL?
(Classical Higher Order Logic/Church's Type Theory)

What is HOL? (Church's Type Theory, Alonzo Church, 1940)

Expressivity	FOL	HOL	Example
Quantification over			
- Individuals	✓	✓	$\forall X.p(f(X))$
- Functions	-	✓	$\forall F.p(F(a))$
- Predicates/Sets/Rels	-	✓	$\forall P.P(f(a))$
Unnamed			
- Functions	-	✓	$(\lambda X.X)$
- Predicates/Sets/Rels	-	✓	$(\lambda X.X \neq a)$
Statements about			
- Functions	-	✓	<i>continuous</i> $(\lambda X.X)$
- Predicates/Sets/Rels	-	✓	<i>reflexive</i> $(=)$
Powerful abbreviations	-	✓	<i>reflexive</i> $= \lambda R. \lambda X. R(X, X)$

What is HOL? (Church's Type Theory, Alonzo Church, 1940)

Expressivity	FOL	HOL	Example
Quantification over			
- Individuals	✓	✓	$\forall X_{\iota}. p_{\iota \rightarrow o}(f_{\iota \rightarrow \iota}(X_{\iota}))$
- Functions	—	✓	$\forall F_{\iota \rightarrow \iota}. p_{\iota \rightarrow o}(F_{\iota \rightarrow o}(a_{\iota}))$
- Predicates/Sets/Rels	—	✓	$\forall P_{\iota \rightarrow o}. P_{\iota \rightarrow o}(f_{\iota \rightarrow \iota}(a_{\iota}))$
Unnamed			
- Functions	—	✓	$(\lambda X_{\iota}. X_{\iota})$
- Predicates/Sets/Rels	—	✓	$(\lambda X_{\iota \rightarrow \iota}. X_{\iota \rightarrow \iota} \neq_{\iota \rightarrow \iota \rightarrow p} a)_{\iota}$
Statements about			
- Functions	—	✓	<i>continuous</i> _{$(\iota \rightarrow \iota) \rightarrow o$} $(\lambda X_{\iota}. X_{\iota})$
- Predicates/Sets/Rels	—	✓	<i>reflexive</i> _{$(\iota \rightarrow \iota \rightarrow o) \rightarrow o$} $(=_{\iota \rightarrow \iota \rightarrow o})$
Powerful abbreviations	—	✓	<i>reflexive</i> _{$(\iota \rightarrow \iota \rightarrow o) \rightarrow o$} $= \lambda R_{(\iota \rightarrow \iota \rightarrow o)}. \lambda X_{\iota}. F$

Simple Types: Prevent Paradoxes and Inconsistencies

- Simple Types

$$\alpha ::= \iota \mid o \mid \alpha_1 \rightarrow \alpha_2$$

- Simple Types

$\alpha ::= \iota \mid o \mid \alpha_1 \rightarrow \alpha_2$

Individuals

Booleans (True and False)

Functions



- Simple Types

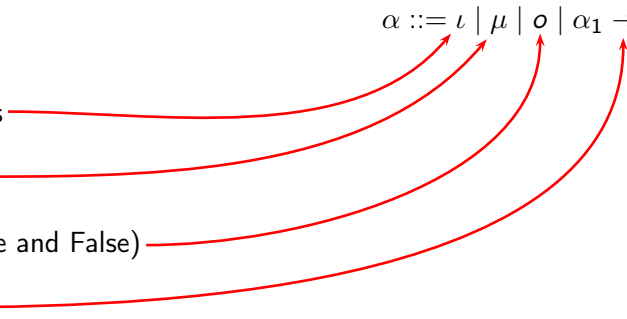
$$\alpha ::= \iota \mid \mu \mid o \mid \alpha_1 \rightarrow \alpha_2$$

Possible worlds

Individuals

Booleans (True and False)

Functions

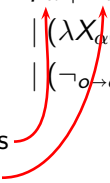


-
- HOL Syntax

$$\begin{aligned}
 s, t \quad ::= & \quad p_\alpha \mid X_\alpha \\
 & \mid ((\lambda X_\alpha. s_\beta)_{\alpha \rightarrow \beta} \mid (s_{\alpha \rightarrow \beta} t_\alpha)_\beta \\
 & \mid (\neg_{o \rightarrow o} s_o)_o \mid (s_o \vee_{o \rightarrow o \rightarrow o} t_o)_o \mid (\forall X_\alpha. t_o)_o
 \end{aligned}$$

Constant Symbols

Variable Symbols



-
- HOL Syntax

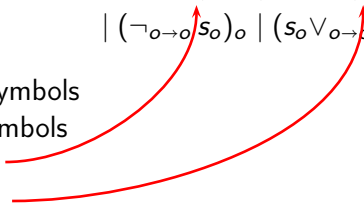
$$\begin{aligned}
 s, t \quad ::= & \quad p_\alpha \mid X_\alpha \\
 & \mid (\lambda X_\alpha. s)_\beta \mid (s_{\alpha \rightarrow \beta} t_\alpha)_\beta \\
 & \mid (\neg_{o \rightarrow o} s_o)_o \mid (s_o \vee_{o \rightarrow o} t_o)_o \mid (\forall X_\alpha. t_o)_o
 \end{aligned}$$

Constant Symbols

Variable Symbols

Abstraction

Application



-
- HOL Syntax

$$\begin{aligned}
 s, t \quad ::= & \quad p_\alpha \mid X_\alpha \\
 & \mid (\lambda X_\alpha. s_\beta)_{\alpha \rightarrow \beta} \mid (s_{\alpha \rightarrow \beta} t_\alpha)_\beta \\
 & \mid (\neg_{o \rightarrow o} s_o)_o \mid (s_o \vee_{o \rightarrow o \rightarrow o} t_o)_o \mid (\forall X_\alpha. t_o)_o
 \end{aligned}$$

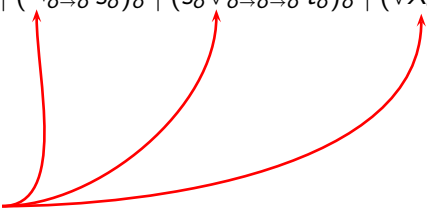
Constant Symbols

Variable Symbols

Abstraction

Application

Logical Connectives



-
- HOL Syntax

$$\begin{aligned}
 s, t \quad ::= & \quad p_\alpha \mid X_\alpha \\
 & \mid (\lambda X_\alpha. s_\beta)_{\alpha \rightarrow \beta} \mid (s_{\alpha \rightarrow \beta} t_\alpha)_\beta \\
 & \mid (\neg_{o \rightarrow o} s_o)_o \mid (s_o \vee_{o \rightarrow o \rightarrow o} t_o)_o \mid \underbrace{(\forall X_\alpha. t_o)_o}_{(\Pi_{(\alpha \rightarrow o) \rightarrow o} (\lambda X_\alpha. t_o))_o}
 \end{aligned}$$

-
- HOL Syntax

$$\begin{aligned}
 s, t \quad ::= \quad & p_\alpha \mid X_\alpha \\
 & \mid (\lambda X_\alpha. s_\beta)_{\alpha \rightarrow \beta} \mid (s_{\alpha \rightarrow \beta} t_\alpha)_\beta \\
 & \mid (\neg_{o \rightarrow o} s_o)_o \mid (s_o \vee_{o \rightarrow o \rightarrow o} t_o)_o \mid (\Pi_{(\alpha \rightarrow o) \rightarrow o} (\lambda X_\alpha. t_o))_o
 \end{aligned}$$

- HOL is (meanwhile) well understood

- Origin

[Church, J.Symb.Log., 1940]

- Henkin-Semantics

[Henkin, J.Symb.Log., 1950]

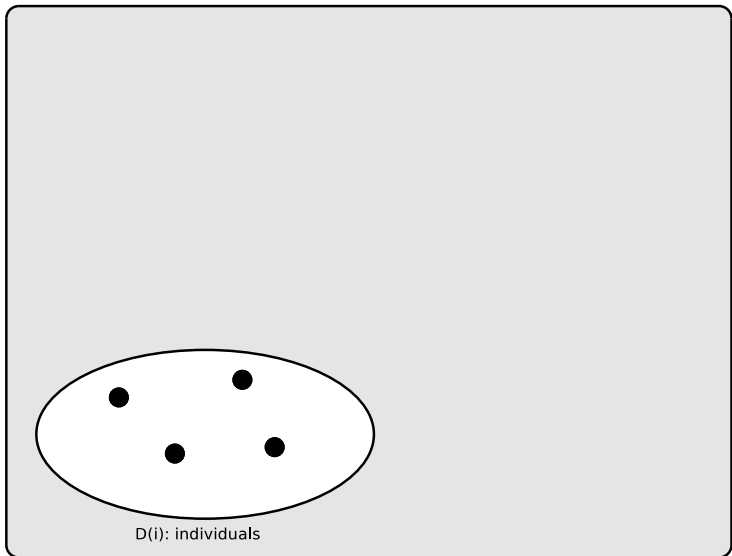
[Andrews, J.Symb.Log., 1971, 1972]

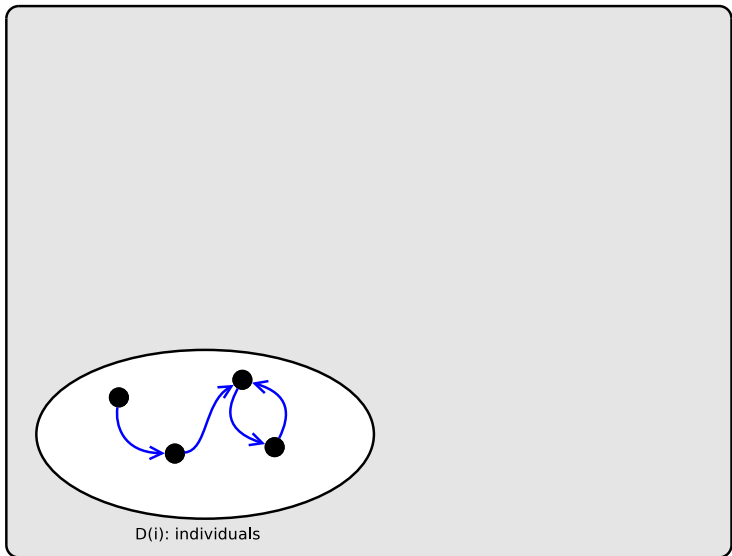
- Extens./Intens.

[Benzmüller et al., J.Symb.Log., 2004]

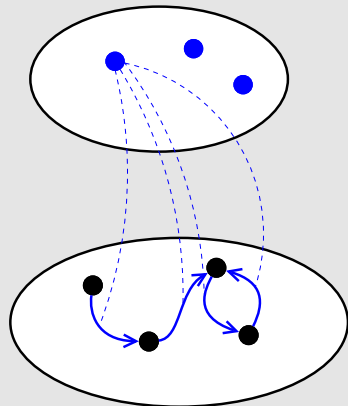
[Muskens, J.Symb.Log., 2007]

- HOL with Henkin-Semantics: **semi-decidable & compact (like FOL)**

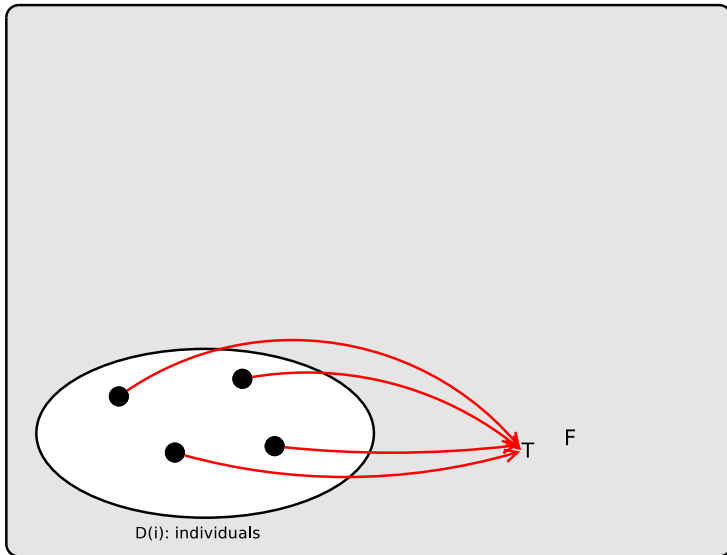


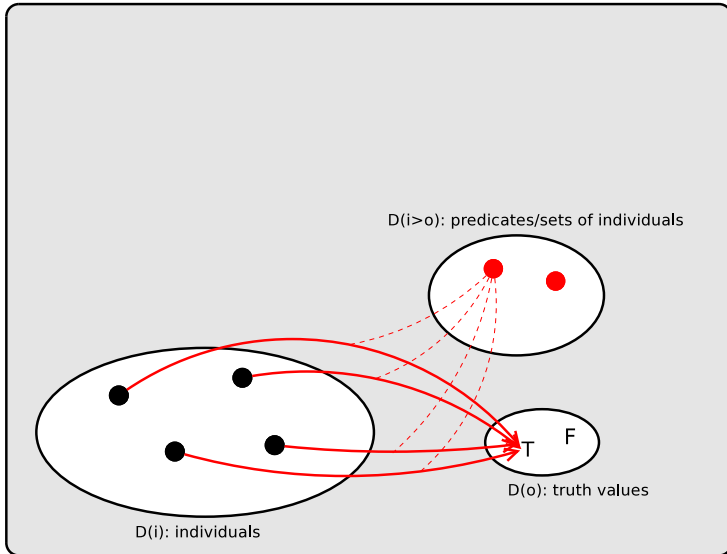


$D(i \rightarrow i)$: functions from indiv. to indiv.

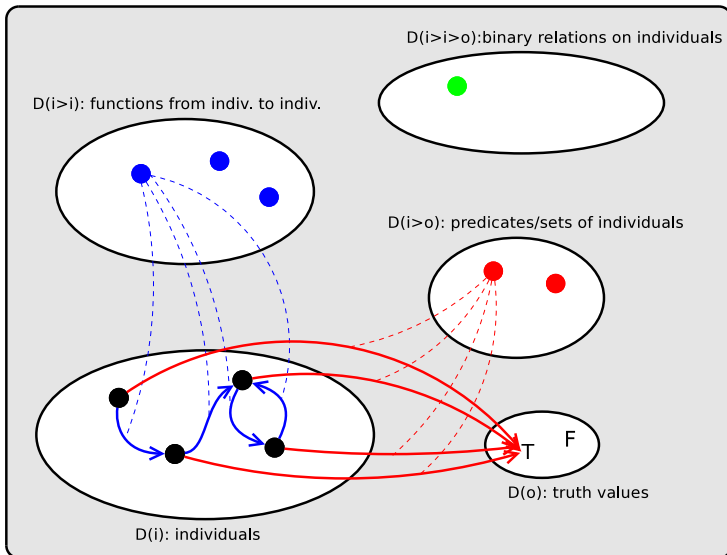


$D(i)$: individuals

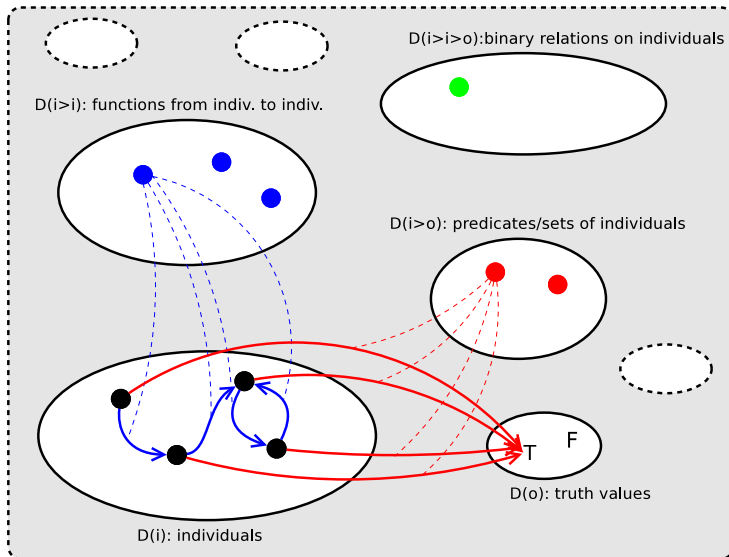




Semantics of HOL



Semantics of HOL



Sets and Relations in HOL

\in	$:=$	$\lambda x. \lambda A. A(x)$	
\emptyset	$:=$	$\lambda x. \perp$	
\cap	$:=$	$\lambda A. \lambda B. (\lambda x. x \in A \wedge x \in B)$	$\{x \mid x \in A \text{ or } x \in B\}$
\cup	$:=$	$\lambda A. \lambda B. (\lambda x. x \in A \vee x \in B)$	
\setminus	$:=$	$\lambda A. \lambda B. (\lambda x. x \in A \vee x \notin B)$	
...			
\subseteq	$:=$	$\lambda A. \lambda B. (\forall x. x \in A \Rightarrow x \in B)$	
\mathcal{P}	$:=$	$\lambda A. (\lambda B. B \subseteq A)$	
...			
reflexive	$:=$	$\lambda R. (\forall x. R(x, x))$	
transitive	$:=$	$\lambda R. (\forall x, y, z. (R(x, y) \wedge R(y, z)) \Rightarrow R(x, z))$	
...			

[BenzmüllerEtAl., Journal of Applied Logic, 2008]

Typed Sets and Relations in HOL

$$\in \quad := \quad \lambda x_{\alpha}. \lambda A_{\alpha \rightarrow o}. A(x)$$

$$\emptyset \quad := \quad \lambda x_{\alpha}. \perp$$

$$\cap \quad := \quad \lambda A_{\alpha \rightarrow o}. \lambda B_{\alpha \rightarrow o}. (\lambda x_{\alpha}. x \in A \wedge x \in B)$$

$$\cup \quad := \quad \lambda A_{\alpha \rightarrow o}. \lambda B_{\alpha \rightarrow o}. (\lambda x_{\alpha}. x \in A \vee x \in B)$$

$$\setminus \quad := \quad \lambda A_{\alpha \rightarrow o}. \lambda B_{\alpha \rightarrow o}. (\lambda x_{\alpha}. x \in A \vee x \notin B)$$

...

Typed Sets and Relations in HOL

$$\begin{aligned}
 \in & \quad := \quad \lambda x_{\alpha}. \lambda A_{\alpha \rightarrow o}. A(x) \\
 \emptyset & \quad := \quad \lambda x_{\alpha}. \perp \\
 \cap & \quad := \quad \lambda A_{\alpha \rightarrow o}. \lambda B_{\alpha \rightarrow o}. (\lambda x_{\alpha}. x \in A \wedge x \in B) \\
 \cup & \quad := \quad \lambda A_{\alpha \rightarrow o}. \lambda B_{\alpha \rightarrow o}. (\lambda x_{\alpha}. x \in A \vee x \in B) \\
 \setminus & \quad := \quad \lambda A_{\alpha \rightarrow o}. \lambda B_{\alpha \rightarrow o}. (\lambda x_{\alpha}. x \in A \vee x \notin B) \\
 \dots &
 \end{aligned}$$

Polymorphism is a Challenge for Automation

- One source of indeterminism / blind guessing

[TheissBenzmüller, IWIL-WS@LPAR, 2006]

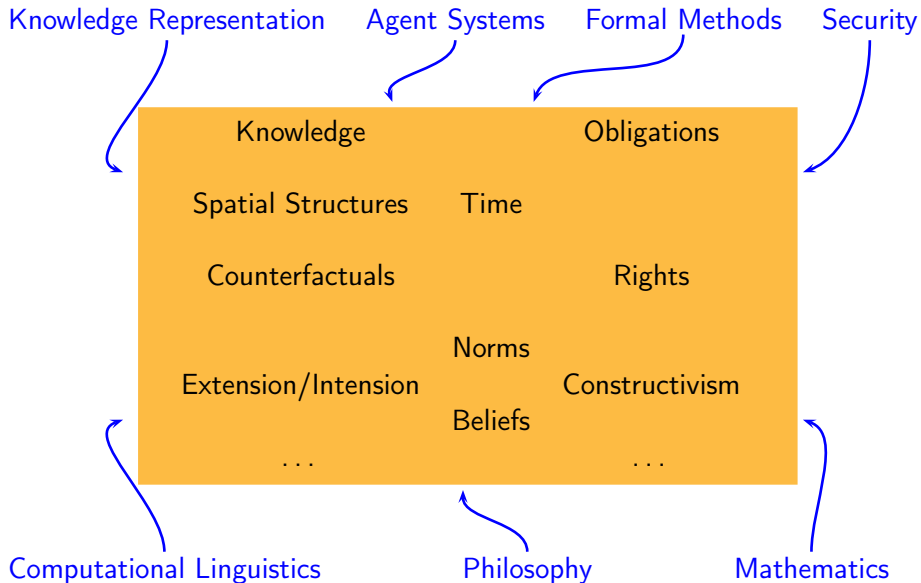


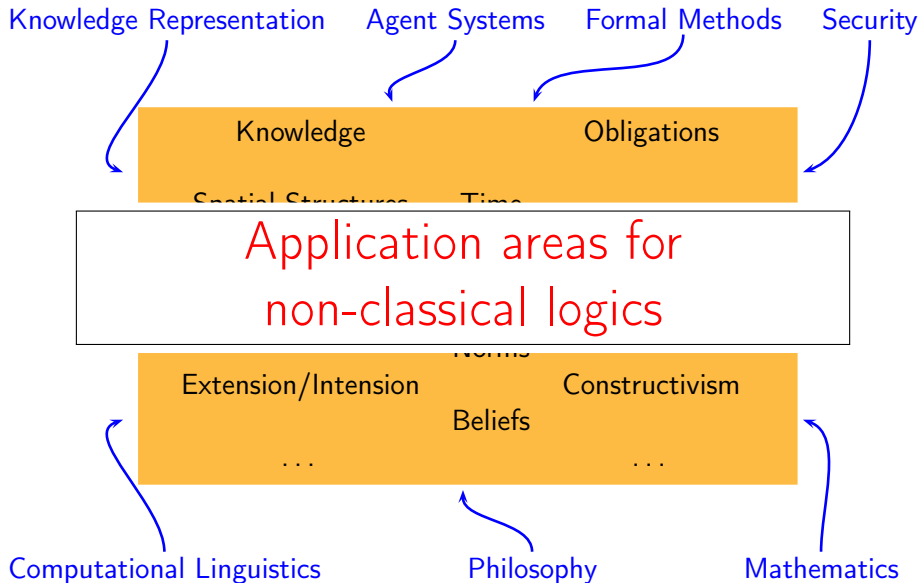
Mechanization & Automation of HOL

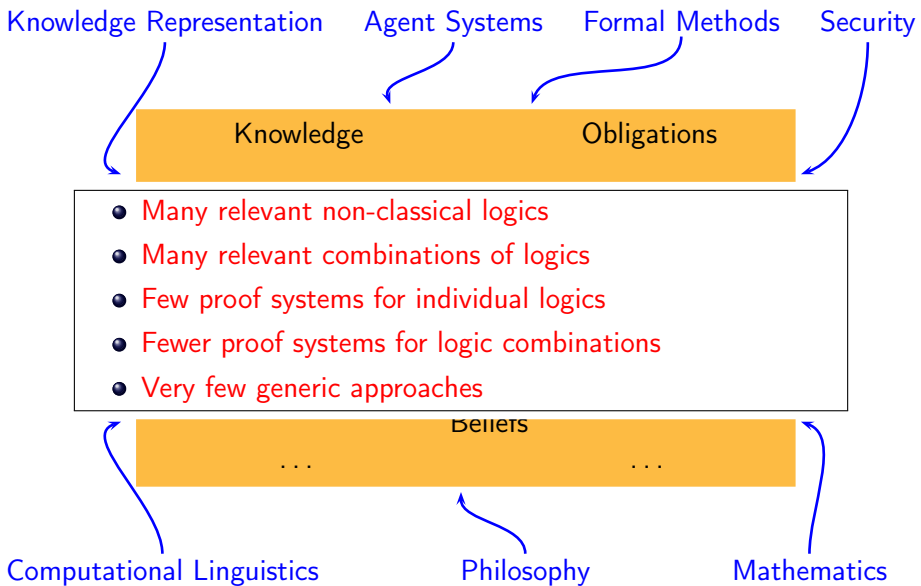
HOL Applications in Formal Methods

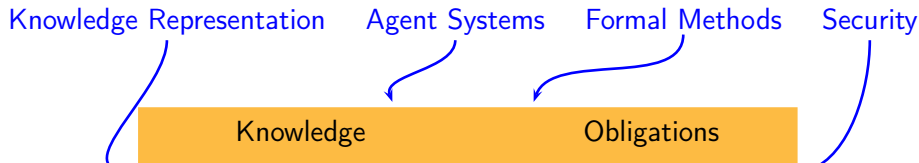
- Systems: ^{Cambridge/München} Isabelle/HOL , ^{Cambridge} HOL4 , ^{INTEL} HOL-Light, ^{SRI} PVS, ^{Cornell} Nuprl, . . . , OMEGA
- Project example (**formal verification**)
 - Flyspeck (Th. Hales, U Pittsburgh)
 - Goal: formal verification of his proof of Kepler's Conjecture (1611)
 - Application of HOL-Light & Isabelle/HOL & . . .
 - 'may take up to 20 work-years' (Flyspeck website)
- Crucial resource: **user interaction**
- Countermeasure: **improving the automation support**





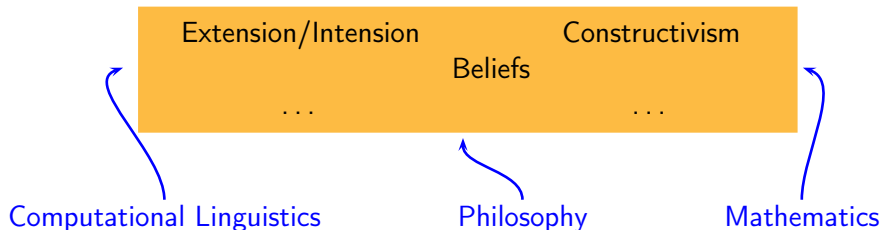






Core Questions:

- 1 Classical Higher Order Logic (HOL) as Universal Logic?
- 2 HOL Provers & Model Finders as Generic Reasoning Tools?
- 3 Combinations with Specialist Reasoners (if available)?



Undecidable and Infinitary Unification

$$\exists F_{l \rightarrow l}. F(g(x)) = g(F(x))$$

- (1) $F \leftarrow \lambda y_i. y$
- (2) $F \leftarrow \lambda y_i. g(y)$
- (3) $F \leftarrow \lambda y_i. g(g(y))$
- (4) ...

→ enforce decidability



Primitive Substitution

Example Theorem:

$$\exists S. reflexive(S)$$

Negation and Expansion of Definitions:

$$\neg \exists S. (\forall x_i. S(x, x))$$

Clause Normalisation ($a(S)$ Skolem term):

$$\neg S(a(S), a(S))$$

Guess some suitable instances for S

$$S \leftarrow \lambda y. \lambda z. \top$$

$$\rightsquigarrow \neg \top$$

$$S \leftarrow \lambda y. \lambda z. V(y, z) = W(y, z)$$

$$\rightsquigarrow V(a(S), a(S)) \neq W(a(S), a(S))$$

$$S \leftarrow \dots$$



Cut rule

$$\frac{A \Rightarrow C \quad C \Rightarrow B}{A \Rightarrow B}$$

considered as bad in ATP

Cut rule

$$\frac{A \Rightarrow C \quad C \Rightarrow B}{A \Rightarrow B}$$

considered as bad in ATP

[IJCAR-06]: Axioms that imply Cut

- Axiom of excluded middle
- Comprehension axioms
- Functional and Boolean extensionality
- Leibniz and other definitions of equality
- Axiom of induction
- Axiom of choice
- Axiom of description

[BenzmüllerEtAl., Logical Methods in Computer Science, 2009]

Cut rule

$$\frac{A \Rightarrow C \quad C \Rightarrow B}{A \Rightarrow B}$$

considered as bad in ATP

Calculi that avoid axioms

- Axiom of excluded middle ✓
- Comprehension axioms ✓
- Functional and Boolean extensionality ✓ [CADE-98, PhD-99]
- Leibniz and other definitions of equality ✓ [CADE-99, PhD-99]
- Axiom of induction ?
- Axiom of choice ✓ (see recent work of Brown)
- Axiom of description ✓ (see recent work of Brown)

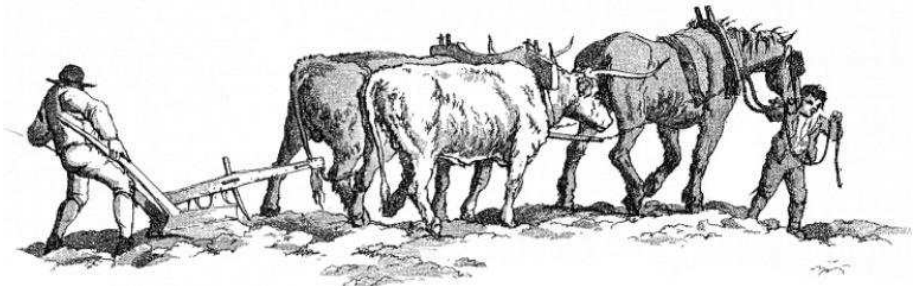
[BenzmüllerEtAl., Logical Methods in Computer Science, 2009]

LEO-II

UNIVERSITY OF
CAMBRIDGE

UNIVERSITÄT
DES
SAARLANDES

An Effective Higher-Order Theorem Prover



LEO-II employs FO-ATPs:

E, Spass, Vampire

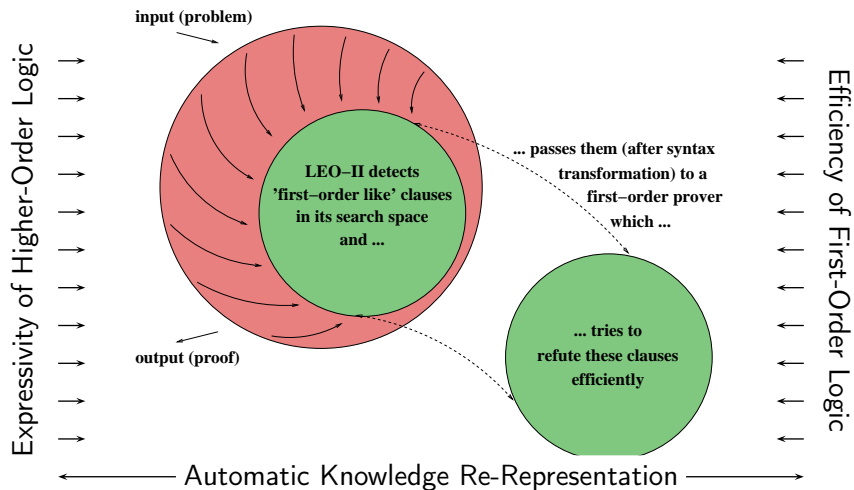
Download and further Information: www.leoprover.org

[BenzmüllerEtAl., IJCAR, 2008]

- TPS system of Peter Andrews et al.
- LEO hardwired to Ω mega (predecessor of LEO-II)
- Agent-based architecture Ω -Ants
(with V. Sorge) [AIMSA-98,EPIA-99,Calculemus-00]
- Collaboration of LEO with FO-ATP via Ω -Ants
(with V. Sorge) [KI-01,LPAR-05,JAL-08]
- Progress in Higher-Order Termination
(with F. Theiss and A. Fietzke) [IWIL-06]

⇒ Development of LEO-II with L. Paulson at Cambridge University

Architecture of LEO-II



Outline of the LEO-II Loop

Main Termination Criterion: generation of empty clause, then raise exception/stop

Outline of the LEO-II Loop

Main Termination Criterion: generation of empty clause, then raise exception/stop

Pre-Processing

- abbreviation expansion, splitting, extensional normalisation and depth-bound extensional pre-unification, Skolemization, primitive substitution, simplification, etc.
- initialize clause sets: passive=emptyset, active=results from above
- call fo-atp with fo-like clauses; stop if refutation found

Outline of the LEO-II Loop

Main Termination Criterion: generation of empty clause, then raise exception/stop
Pre-Processing

- abbreviation expansion, splitting, extensional normalisation and depth-bound extensional pre-unification, Skolemization, primitive substitution, simplification, etc.
- initialize clause sets: passive=emptyset, active=results from above
- call fo-atp with fo-like clauses; stop if refutation found

LEO-II Loop

- while 'Reasoning-Timeout' not yet reached do
 - increment loop counter (stop when maximal number of loops reached)
 - call fo-atp with fo-like clauses; stop if refutation found
 - choose new lightest clause from active and rename free vars
 - if lightest is-subsumed-by passive then nothing else
 - remove subsumed clauses from active and add lightest clause to passive
 - resolve all clauses in active against lightest clauses
 - (apply primitive substitution to lightest clause)
 - (apply positive boolean extensionality to lightest clause)
 - apply restricted factorization to lightest clause
 - process resulting clauses with: extensional normalisation and depth-bound extensional pre-unification, simplification
 - add resulting clauses to active

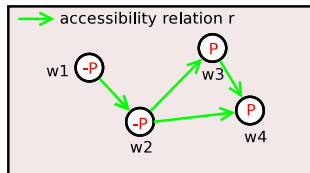
```
(** The Main Loop *)
```

```
let loop (st:state) =  
  while (not (check_local_max_time st))  
  do  
    let lc = inc_loop_count st in  
    if (st.flags.max_loop_count > 0) && (st.loop_count >= st.flags.max_loop_count)  
    then raise (Failure "Max loops") else ();  
    if (not (st.flags.atp_prover = "none"))  
    then call_fo_atp_according_to_frequency_flag st st.flags.atp_prover  
    else ();  
    let lightest = choose_and_remove_lightest_from_active st in  
    let lightest' = rename_free_variables lightest st in  
    if is_subsumed_by lightest' (Clauseset.elements st.passive) st "fo-match" then ()  
    else  
      (  
        set_passive st (list_to_set (delete_subsumed_clauses (Clauseset.elements st.passive)  
                                                                lightest' st "fo-match"));  
        add_to_passive st lightest';  
        let res_resolve = List.fold_right (fun cl cll -> (resolve lightest cl st)@ccll  
                                           (Clauseset.elements st.passive) [] in  
          let res_prim_subst = [] and res_pos_bool = []  
          and res_fac_restr = (raise_to_list factorize_restricted) [lightest] st in  
          let res_processed =  
            compose [(raise_to_list unify_pre_ext);  
                    exhaustive (raise_to_list cnf_normalize_step);  
                    exhaustive (raise_to_list simplify)]  
                    (res_resolve@res_prim_subst@res_pos_bool@res_fac_restr) st in  
            index_clause_list_with_role res_processed st;  
            set_active st (list_to_set (res_processed@(Clauseset.elements st.active)));  
          )  
      )  
  done
```

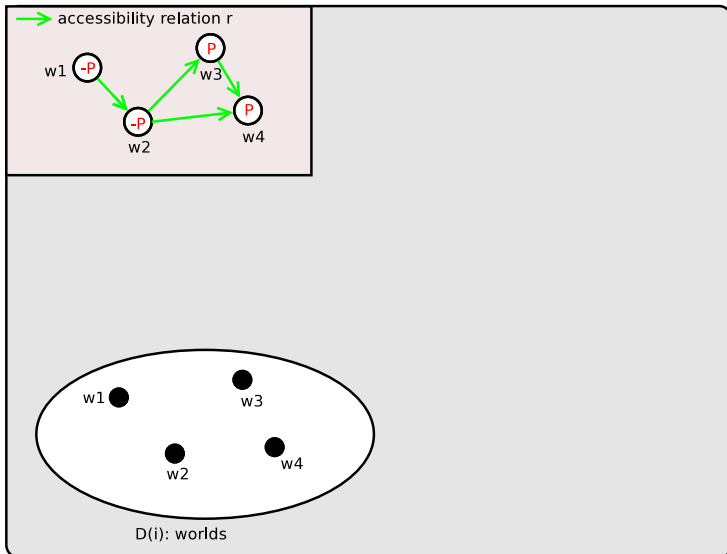


Examples of Natural Fragments of HOL:
Quantified Multimodal Logics & Others

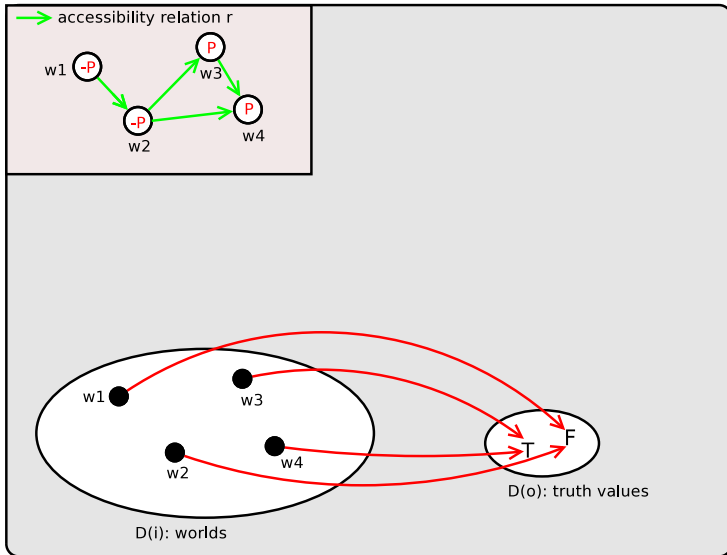
Combining the Kripke View and the Tarski View on Logics



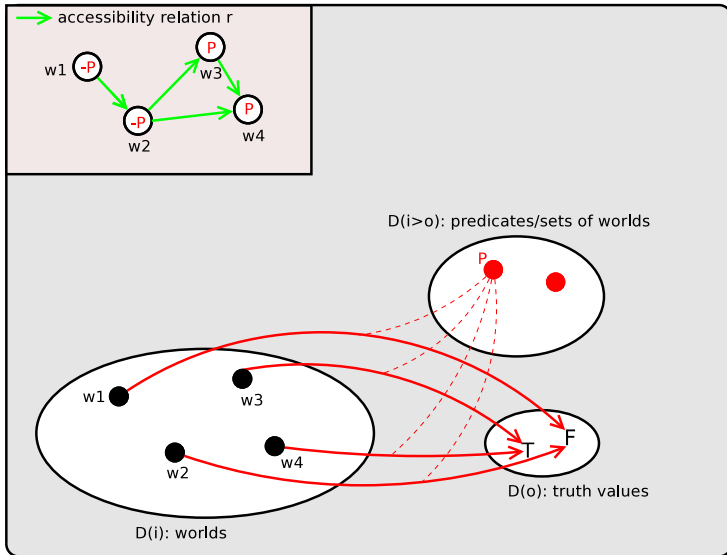
Combining the Kripke View and the Tarski View on Logics



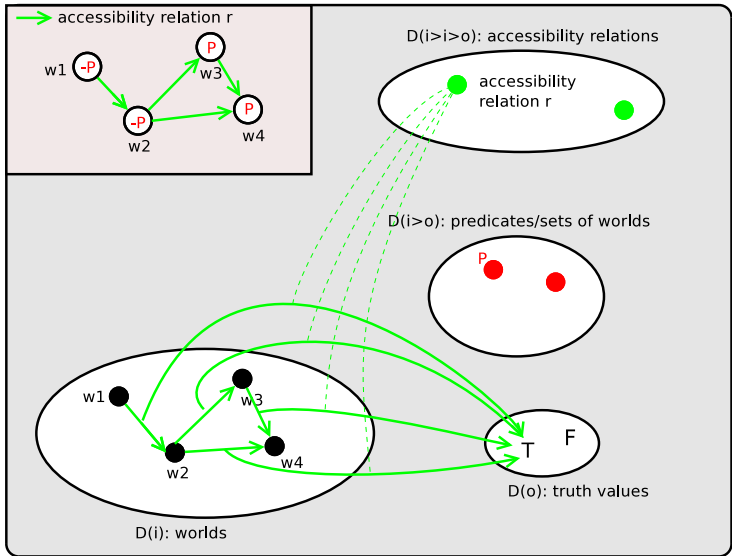
Combining the Kripke View and the Tarski View on Logics



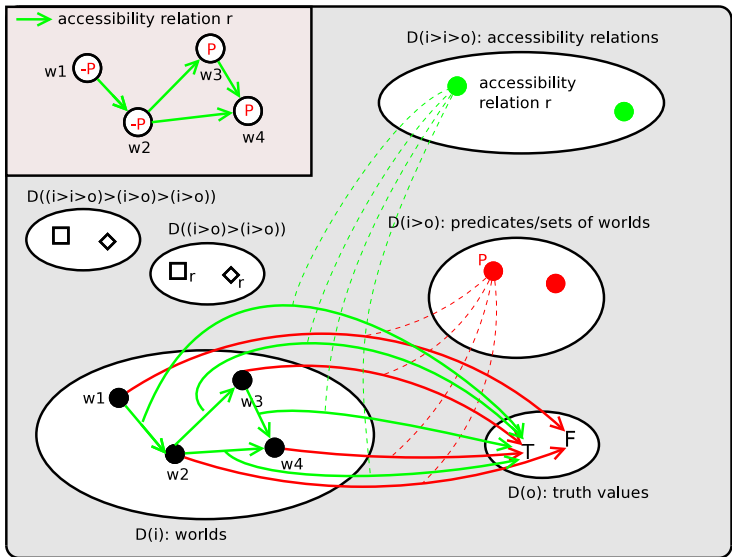
Combining the Kripke View and the Tarski View on Logics



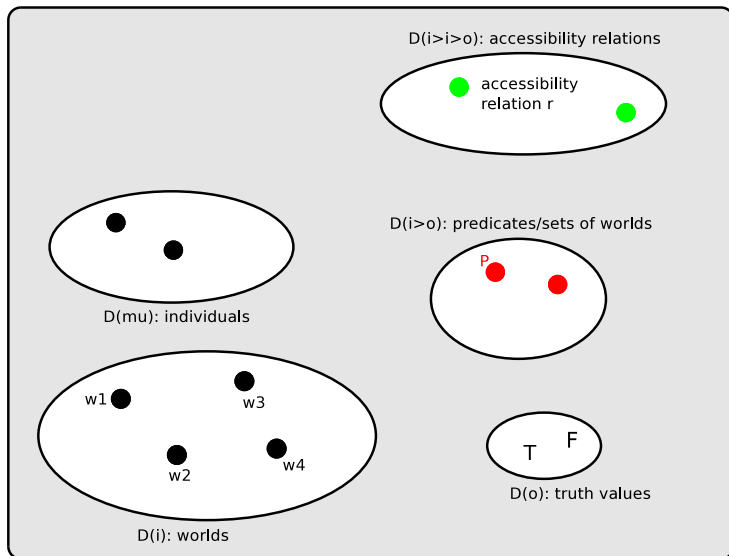
Combining the Kripke View and the Tarski View on Logics



Combining the Kripke View and the Tarski View on Logics



Combining the Kripke View and the Tarski View on Logics



- Syntax (MML):

$$s, t ::= P \mid \neg s \mid s \vee t \mid \Box_r s$$

-

Syntax MML

- formulas s

Kripke Semantics

- worlds w

- accessibility relations r

explicit
→
transformation

First Order Logic

e.g. work of Ohlbach

Not Needed!

- Syntax (MML):

$$s, t ::= P \mid \neg s \mid s \vee t \mid \Box_r s$$

HOL

Syntax MML

- formulas s

Kripke Semantics

- worlds w

- accessibility relations r

→ terms $s_{L \rightarrow o}$

→ terms w_L

→ terms $r_{L \rightarrow L \rightarrow o}$

- Syntax (MML): $s, t ::= P \mid \neg s \mid s \vee t \mid \Box_r s$

HOL

Syntax MML

- formulas s

Kripke Semantics

- worlds w

- accessibility relations r

→ terms $s_{\iota \rightarrow o}$

→ terms w_ι

→ terms $r_{\iota \rightarrow \iota \rightarrow o}$

- MML Syntax as Abbreviations of HOL-Terms

$$P = \lambda W_\iota. (P_{\iota \rightarrow o} W) = P_{\iota \rightarrow o}$$

$$\neg = \lambda S_{\iota \rightarrow o}. \lambda W_\iota. \neg(S W)$$

$$\vee = \lambda S_{\iota \rightarrow o}. \lambda T_{\iota \rightarrow o}. \lambda W_\iota. (S W) \vee (T W)$$

$$\Box = \lambda R_{\iota \rightarrow \iota \rightarrow o}. \lambda S_{\iota \rightarrow o}. \lambda W_\iota. \forall V_\iota. \neg(R W V) \vee (S V)$$

[BenzmüllerPaulson, Log.J.IGPL, 2010], [BenzmüllerPaulson, Logica Universalis, to appear]

- Syntax (MML):

$$s, t ::= P \mid \neg s \mid s \vee t \mid \Box_r s$$

HOL

Syntax MML

- formulas s

Kripke Semantics

- worlds w

- accessibility relations r

→ terms $s_{\iota \rightarrow o}$

→ terms w_ι

→ terms $r_{\iota \rightarrow \iota \rightarrow o}$

- MML Syntax as Abbreviations of HOL-Terms

$$P = \lambda W_\iota. (P_{\iota \rightarrow o} W) = P_{\iota \rightarrow o}$$

$$\neg = \lambda S_{\iota \rightarrow o}. \lambda W_\iota. \neg(S W)$$

$$\vee = \lambda S_{\iota \rightarrow o}. \lambda T_{\iota \rightarrow o}. \lambda W_\iota. (S W) \vee (T W)$$

$$\Box = \lambda R_{\iota \rightarrow \iota \rightarrow o}. \lambda S_{\iota \rightarrow o}. \lambda W_\iota. \forall V_\iota. \neg(R W V) \vee (S V)$$

$$(\forall^P), \forall^\mu = \lambda Q_{\mu \rightarrow (\iota \rightarrow o)}. \lambda W_\iota. \forall X_\mu. (Q X W)$$

- Syntax (MML):

$$s, t ::= P \mid \neg s \mid s \vee t \mid \Box_r s$$

HOL

Syntax MML

- formulas s

Kripke Semantics

- worlds w

- accessibility relations r

- terms $s_{\iota \rightarrow o}$

- terms w_ι

- terms $r_{\iota \rightarrow \iota \rightarrow o}$

- MML Syntax as Abbreviations of HOL-Terms

$$P = \lambda W_\iota. (P_{\iota \rightarrow o} W) = P_{\iota \rightarrow o}$$

$$\neg = \lambda S_{\iota \rightarrow o}. \lambda W_\iota. \neg(S W)$$

$$\vee = \lambda S_{\iota \rightarrow o}. \lambda T_{\iota \rightarrow o}. \lambda W_\iota. (S W) \vee (T W)$$

$$\Box = \lambda R_{\iota \rightarrow \iota \rightarrow o}. \lambda S_{\iota \rightarrow o}. \lambda W_\iota. \forall V_\iota. \neg(R W V) \vee (S V)$$

$$(\forall P), \forall^\mu = \lambda Q_{\mu \rightarrow (\iota \rightarrow o)}. \lambda W_\iota. \forall X_\mu. (Q X W)$$

$$s \Rightarrow t = \lambda S_{\iota \rightarrow o}. \lambda T_{\iota \rightarrow o}. \lambda W_\iota. \forall V_\iota. \neg(f W S V) \vee (T V)$$

[BenzmüllerGenovese, NCMPL, 2011], [BenzmüllerGabbayGenoveseRispoli, Logica Universalis, to appear]

- Validity

$$\text{valid} = \lambda\phi_{\iota \rightarrow o}. \forall W_{\iota}. \phi W$$

Also

- Satisfiability
- Countersatisfiability
- Unsatisfiability

- Kripke style semantics

$M, w \models P$	arbitrary	
$M, w \models \neg s$	iff	not $M, w \models s$
$M, w \models s \vee t$	iff	$M, w \models s$ or $M, w \models t$
$M, w \models \Box_r s$	iff	$M, v \models s$ for all v mit $r(w, v)$

- Semantic embedding: $ML \longrightarrow HOL$ terms of type $\iota \rightarrow o$
 Base type ι is identified with set of worlds $W \neq \emptyset$

$$P = \lambda W_{\iota}. (P_{\iota \rightarrow o} W) = P_{\iota \rightarrow o}$$

$$\neg = \lambda S_{\iota \rightarrow o}. \lambda W_{\iota}. \neg(S W)$$

$$\vee = \lambda S_{\iota \rightarrow o}. \lambda T_{\iota \rightarrow o}. \lambda W_{\iota}. (S W) \vee (T W)$$

$$\Box = \lambda R_{\iota \rightarrow \iota \rightarrow o}. \lambda S_{\iota \rightarrow o}. \lambda W_{\iota}. \forall V_{\iota}. \neg(R W V) \vee (S V)$$

- Kripke style semantics

$M, w \models P$	arbitrary	
$M, w \models \neg s$	iff	not $M, w \models s$
$M, w \models s \vee t$	iff	$M, w \models s$ or $M, w \models t$
$M, w \models \Box_r s$	iff	$M, v \models s$ for all v mit $r(w, v)$

- Semantic embedding: $ML \longrightarrow HOL$ terms of type $\iota \rightarrow o$
 Base type ι is identified with set of worlds $W \neq \emptyset$

$$P = \lambda W_{\iota}. (P_{\iota \rightarrow o} W) = P_{\iota \rightarrow o}$$

$$\neg = \lambda S_{\iota \rightarrow o}. \lambda W_{\iota}. \neg(S W)$$

$$\vee = \lambda S_{\iota \rightarrow o}. \lambda T_{\iota \rightarrow o}. \lambda W_{\iota}. (S W) \vee (T W)$$

$$\Box = \lambda R_{\iota \rightarrow \iota \rightarrow o}. \lambda S_{\iota \rightarrow o}. \lambda W_{\iota}. \forall V_{\iota}. \neg(R W V) \vee (S V)$$

- Kripke style semantics

$M, w \models P$	arbitrary	
$M, w \models \neg s$	iff	not $M, w \models s$
$M, w \models s \vee t$	iff	$M, w \models s$ or $M, w \models t$
$M, w \models \Box_r s$	iff	$M, v \models s$ for all v mit $r(w, v)$

- Semantic embedding: $ML \longrightarrow HOL$ terms of type $\iota \rightarrow o$
 Base type ι is identified with set of worlds $W \neq \emptyset$

$$\begin{aligned}
 P &= \lambda W_{\iota}. (P_{\iota \rightarrow o} W) = P_{\iota \rightarrow o} \\
 \neg &= \lambda S_{\iota \rightarrow o}. \lambda W_{\iota}. \neg(S W) \\
 \vee s &= \lambda T_{\iota \rightarrow o}. \lambda W_{\iota}. (s W) \vee (T W) \\
 \Box &= \lambda R_{\iota \rightarrow \iota \rightarrow o}. \lambda S_{\iota \rightarrow o}. \lambda W_{\iota}. \forall V_{\iota}. \neg(R W V) \vee (S V)
 \end{aligned}$$

- Kripke style semantics

$M, w \models P$	arbitrary	
$M, w \models \neg s$	iff	not $M, w \models s$
$M, w \models s \vee t$	iff	$M, w \models s$ or $M, w \models t$
$M, w \models \Box_r s$	iff	$M, v \models s$ for all v mit $r(w, v)$

- Semantic embedding: $ML \longrightarrow HOL$ terms of type $\iota \rightarrow o$
 Base type ι is identified with set of worlds $W \neq \emptyset$

$$\begin{aligned}
 P &= \lambda W_\iota. (P_{\iota \rightarrow o} W) = P_{\iota \rightarrow o} \\
 \neg &= \lambda S_{\iota \rightarrow o}. \lambda W_\iota. \neg(S W) \\
 \vee s t &= \lambda W_\iota. (s W) \vee (t W) \\
 \Box &= \lambda R_{\iota \rightarrow \iota \rightarrow o}. \lambda S_{\iota \rightarrow o}. \lambda W_\iota. \forall V_\iota. \neg(R W V) \vee (S V)
 \end{aligned}$$

- Kripke style semantics

$M, w \models P$	arbitrary	
$M, w \models \neg s$	iff	not $M, w \models s$
$M, w \models s \vee t$	iff	$M, w \models s$ or $M, w \models t$
$M, w \models \Box_r s$	iff	$M, v \models s$ for all v mit $r(w, v)$

- Semantic embedding: $ML \longrightarrow HOL$ terms of type $\iota \rightarrow o$
 Base type ι is identified with set of worlds $W \neq \emptyset$

$$\begin{aligned}
 P &= \lambda W_{\iota}. (P_{\iota \rightarrow o} W) = P_{\iota \rightarrow o} \\
 \neg &= \lambda S_{\iota \rightarrow o}. \lambda W_{\iota}. \neg(S W) \\
 (\vee s t) W &= (s W) \vee (t W) \\
 \Box &= \lambda R_{\iota \rightarrow \iota \rightarrow o}. \lambda S_{\iota \rightarrow o}. \lambda W_{\iota}. \forall V_{\iota}. \neg(R W V) \vee (S V)
 \end{aligned}$$

- Kripke style semantics

$M, w \models P$	arbitrary	
$M, w \models \neg s$	iff	not $M, w \models s$
$M, w \models s \vee t$	iff	$M, w \models s$ or $M, w \models t$
$M, w \models \Box_r s$	iff	$M, v \models s$ for all v mit $r(w, v)$

- Semantic embedding: $ML \longrightarrow HOL$ terms of type $\iota \rightarrow o$
 Base type ι is identified with set of worlds $W \neq \emptyset$

$$P = \lambda W_{\iota}. (P_{\iota \rightarrow o} W) = P_{\iota \rightarrow o}$$

$$\neg = \lambda S_{\iota \rightarrow o}. \lambda W_{\iota}. \neg(S W)$$

$$\vee = \lambda S_{\iota \rightarrow o}. \lambda T_{\iota \rightarrow o}. \lambda W_{\iota}. (S W) \vee (T W)$$

$$\Box = \lambda R_{\iota \rightarrow \iota \rightarrow o}. \lambda S_{\iota \rightarrow o}. \lambda W_{\iota}. \forall V_{\iota}. \neg(R W V) \vee (S V)$$

- Kripke style semantics

$M, w \models P$	arbitrary	
$M, w \models \neg s$	iff	not $M, w \models s$
$M, w \models s \vee t$	iff	$M, w \models s$ or $M, w \models t$
$M, w \models \Box_r s$	iff	$M, v \models s$ for all v mit $r(w, v)$

- Semantic embedding: $ML \longrightarrow HOL$ terms of type $\iota \rightarrow o$
 Base type ι is identified with set of worlds $W \neq \emptyset$

$$P = \lambda W_{\iota}. (P_{\iota \rightarrow o} W) = P_{\iota \rightarrow o}$$

$$\neg = \lambda S_{\iota \rightarrow o}. \lambda W_{\iota}. \neg(S W)$$

$$\vee = \lambda S_{\iota \rightarrow o}. \lambda T_{\iota \rightarrow o}. \lambda W_{\iota}. (S W) \vee (T W)$$

$$\Box = \lambda R_{\iota \rightarrow \iota \rightarrow o}. \lambda S_{\iota \rightarrow o}. \lambda W_{\iota}. \forall V_{\iota}. \neg(R W V) \vee (S V)$$

- Kripke style semantics

$M, w \models P$	arbitrary	
$M, w \models \neg s$	iff	not $M, w \models s$
$M, w \models s \vee t$	iff	$M, w \models s$ or $M, w \models t$
$M, w \models \Box_r s$	iff	$M, v \models s$ for all v mit $r(w, v)$

- Semantic embedding: $ML \longrightarrow HOL$ terms of type $\iota \rightarrow o$
 Base type ι is identified with set of worlds $W \neq \emptyset$

$$\begin{aligned}
 P &= \lambda W_\iota. (P_{\iota \rightarrow o} W) = P_{\iota \rightarrow o} \\
 \neg &= \lambda S_{\iota \rightarrow o}. \lambda W_\iota. \neg(S W) \\
 \vee &= \lambda S_{\iota \rightarrow o}. \lambda T_{\iota \rightarrow o}. \lambda W_\iota. (S W) \vee (T W) \\
 \Box_r &= \lambda S_{\iota \rightarrow o}. \lambda W_\iota. \forall V_\iota. \neg(r W V) \vee (S V)
 \end{aligned}$$

- Kripke style semantics

higher-order selection function!

$M, w \models P$ arbitrary

$M, w \models \neg s$ iff

not $M, w \models s$

$M, w \models s \vee t$ iff

$M, w \models s$ or $M, w \models t$

$M, w \models \Box_r s$ iff

$M, u \models s$ for all v mit $r(w, v)$

$M, w \models s \xrightarrow{\text{cond}} t$ iff

$M, v \models t$ for all $v \in f(w, [s])$
with $[s] = \{u \mid M, u \models s\}$

- Semantic embedding:

ML \longrightarrow HOL terms of type $\iota \rightarrow o$

Base type ι is identified with set of worlds $W \neq \emptyset$

$P = \lambda W_\iota. (P_{\iota \rightarrow o} W) = P_{\iota \rightarrow o}$

$\neg = \lambda S_{\iota \rightarrow o}. \lambda W_\iota. \neg(S W)$

$\vee = \lambda S_{\iota \rightarrow o}. \lambda T_{\iota \rightarrow o}. \lambda W_\iota. (S W) \vee (T W)$

$\Box = \lambda R_{\iota \rightarrow \iota \rightarrow o}. \lambda S_{\iota \rightarrow o}. \lambda W_\iota. \forall V_\iota. \neg(R W V) \vee (S V)$

$\xrightarrow{\text{cond}} = \lambda S_{\iota \rightarrow o}. \lambda T_{\iota \rightarrow o}. \lambda W_\iota. \forall V_\iota. \neg(f W S V) \vee (T V)$

- Kripke style semantics

$M, w \models P$	arbitrary	
$M, w \models \neg s$	iff	not $M, w \models s$
$M, w \models s \vee t$	iff	$M, w \models s$ or $M, w \models t$
$M, w \models \Box_r s$	iff	$M, v \models s$ for all v mit $r(w, v)$
$M, w \models s \overset{cond}{\Rightarrow} t$	iff	$M, v \models t$ for all $v \in f(w, [s])$ with $[s] = \{u \mid M, u \models s\}$

higher-order selection function!

- Semantic embedding:

Base type ι is identified with set of worlds $W \neq \emptyset$

ML \longrightarrow HOL terms of type $\iota \rightarrow o$

P	$=$	$\lambda W_\iota. (P_{\iota \rightarrow o} W) = P_{\iota \rightarrow o}$
\neg	$=$	$\lambda S_{\iota \rightarrow o}. \lambda W_\iota. \neg(S W)$
\vee	$=$	$\lambda S_{\iota \rightarrow o}. \lambda T_{\iota \rightarrow o}. \lambda W_\iota. (S W) \vee (T W)$
\Box	$=$	$\lambda R_{\iota \rightarrow \iota \rightarrow o}. \lambda S_{\iota \rightarrow o}. \lambda W_\iota. \forall V_\iota. \neg(R W V) \vee (S V)$
$\overset{cond}{\Rightarrow}$	$=$	$\lambda S_{\iota \rightarrow o}. \lambda T_{\iota \rightarrow o}. \lambda W_\iota. \forall V_\iota. \neg(f W S V) \vee (T V)$

- Kripke style semantics

higher-order selection function!

$M, w \models P$	arbitrary	
$M, w \models \neg s$	iff	not $M, w \models s$
$M, w \models s \vee t$	iff	$M, w \models s$ or $M, w \models t$
$M, w \models \Box_r s$	iff	$M, v \models s$ for all v mit $r(w, v)$
$M, w \models s \xrightarrow{\text{cond}} t$	iff	$M, v \models t$ for all $v \in f(w, [s])$ with $[s] = \{u \mid M, u \models s\}$

- Semantic embedding:

ML \longrightarrow HOL terms of type $\iota \rightarrow o$

Base type ι is identified with set of worlds $W \neq \emptyset$

$$\begin{aligned}
 P &= \lambda W_\iota. (P_{\iota \rightarrow o} W) = P_{\iota \rightarrow o} \\
 \neg &= \lambda S_{\iota \rightarrow o}. \lambda W_\iota. \neg(S W) \\
 \vee &= \lambda S_{\iota \rightarrow o}. \lambda T_{\iota \rightarrow o}. \lambda W_\iota. (S W) \vee (T W) \\
 \Box &= \lambda R_{\iota \rightarrow \iota \rightarrow o}. \lambda S_{\iota \rightarrow o}. \lambda W_\iota. \forall V_\iota. \neg(R W V) \vee (S V) \\
 \forall^\mu &= \lambda Q_{\mu \rightarrow (\iota \rightarrow o)}. \lambda W_\iota. \forall X_\mu. (Q X W) \\
 \forall^P &= \lambda Q_{(\iota \rightarrow o) \rightarrow (\iota \rightarrow o)}. \lambda W_\iota. \forall P_{\iota \rightarrow o}. (Q P W)
 \end{aligned}$$

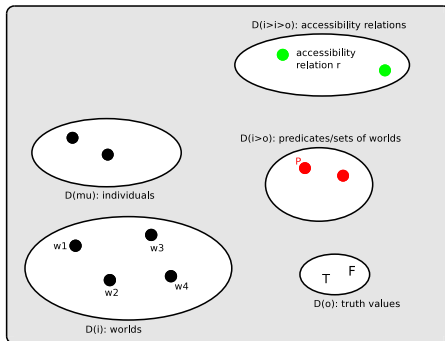
- Remember

$$\text{valid} = \lambda\phi_{t \rightarrow o}. \forall W_t. \phi W$$

Examples on blackboard

- $\text{valid } \forall^\mu X. (m X)$
- $\text{valid } \forall^\mu X. \Box_r (m X)$

Constant Domain

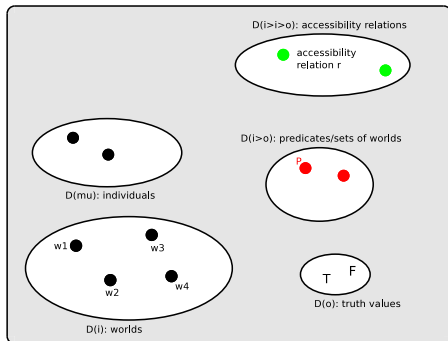


$$\forall^\mu = \lambda Q. \lambda W_L. \forall X_{\mu} (Q X W)$$

[BenzmüllerPaulson, Logica Universalis, to appear]

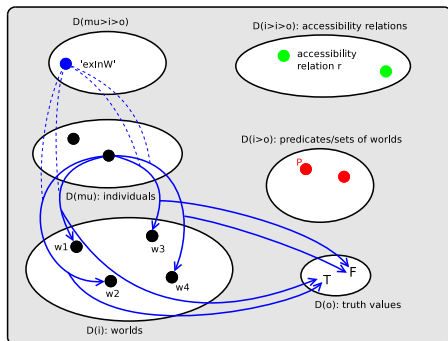
Quantified Modal Logics: Constant versus Cumulative Domain

Constant Domain



$$\forall^\mu = \lambda Q. \lambda W_L. \lambda W_U. \forall X_{\mu}. (Q \ X \ W)$$

Cumulative Domain



$$\forall^\mu = \lambda Q. \lambda W_L. \lambda W_U. \forall X_{\mu}. (\text{exIn}W \ X \ W) \Rightarrow (Q \ X \ W)$$

- 1: $\forall X_{\mu}, V_L, W_L. (\text{exIn}W \ X \ V) \wedge (r \ V \ W) \Rightarrow (\text{exIn}W \ X \ W)$
- 2: $\forall W_L. \exists X_{\mu}. (\text{exIn}W \ X \ W)$
- 3(c): $\forall W_L. (\text{exIn}W \ c \ W)$

[BenzmüllerPaulson, Logica Universalis, to appear]

[ongoing work with Otten and Raths]

Region Connection Calculus (RCC) is a Fragment of HOL

Region Connection Calculus for spatial reasoning [RandellCuiCohn, 1992]

disconnected :	<i>dc</i>	$= \lambda X_{\tau}. \lambda Y_{\tau}. \neg(c X Y)$
part of :	<i>p</i>	$= \lambda X_{\tau}. \lambda Y_{\tau}. \forall Z. ((c Z X) \Rightarrow (c Z Y))$
identical with :	<i>eq</i>	$= \lambda X_{\tau}. \lambda Y_{\tau}. ((p X Y) \wedge (p Y X))$
overlaps :	<i>o</i>	$= \lambda X_{\tau}. \lambda Y_{\tau}. \exists Z. ((p Z X) \wedge (p Z Y))$
partially o :	<i>po</i>	$= \lambda X_{\tau}. \lambda Y_{\tau}. ((o X Y) \wedge \neg(p X Y) \wedge \neg(p Y X))$
ext. connected :	<i>ec</i>	$= \lambda X_{\tau}. \lambda Y_{\tau}. ((c X Y) \wedge \neg(o X Y))$
proper part :	<i>pp</i>	$= \lambda X_{\tau}. \lambda Y_{\tau}. ((p X Y) \wedge \neg(p Y X))$
tangential pp :	<i>tpp</i>	$= \lambda X_{\tau}. \lambda Y_{\tau}. ((pp X Y) \wedge \exists Z. ((ec Z X) \wedge (ec Z Y)))$
nontang. pp :	<i>ntpp</i>	$= \lambda X_{\tau}. \lambda Y_{\tau}. ((pp X Y) \wedge \neg \exists Z. ((ec Z X) \wedge (ec Z Y)))$

[Benzmüller, AMAI, 2011]

Semantic Web Language OWL is a Fragment of HOL

- Class expressions become terms of type $\iota \rightarrow o$
- Class membership becomes class application $(C a)$
- Role expressions become terms of type $\iota \rightarrow \iota \rightarrow o$
- Role membership becomes role application $(R a b)$

Semantic Web Language OWL is a Fragment of HOL

- Class expressions become terms of type $\iota \rightarrow o$
- Class membership becomes class application ($C a$)
- Role expressions become terms of type $\iota \rightarrow \iota \rightarrow o$
- Role membership becomes role application ($R a b$)
- The class connectives $\perp, \top, \neg, \sqcup, \forall, \geq_n$ can be defined as

$$\top = \lambda X_{\iota}. \top$$

$$\perp = \lambda X_{\iota}. \perp$$

$$\neg = \lambda C_{\iota \rightarrow o}. \lambda X_{\iota}. \neg(C X)$$

$$\sqcup = \lambda C_{\iota \rightarrow o}. \lambda D_{\iota \rightarrow o}. \lambda X_{\iota}. (C X) \vee (D X)$$

$$\forall = \lambda R_{\iota \rightarrow \iota \rightarrow o}. \lambda C_{\iota \rightarrow o}. \lambda X_{\iota}. \forall Y_{\iota}. (R X Y) \Rightarrow (C Y)$$

$$\geq_n = \lambda R_{\iota \rightarrow \iota \rightarrow o}. \lambda C_{\iota \rightarrow o}. \lambda X_{\iota}. ((\# \lambda Y_{\iota}. (R X Y) \wedge (C Y)) \geq n)$$

($\#$ is an appropriately defined set cardinality function)

- Role inverse R^{-} is defined as $^{-} = \lambda R_{\iota \rightarrow \iota \rightarrow o}. \lambda X_{\iota}. \lambda Y_{\iota}. (R Y X)$

- Class expressions become terms of type $\iota \rightarrow o$
- Class membership becomes class application ($C a$)
- Role expressions become terms of type $\iota \rightarrow \iota \rightarrow o$
- Role membership becomes role application ($R a b$)
- Definition of further connectives

$$\sqsubseteq^1 = \lambda C_{\iota \rightarrow o}. \lambda D_{\iota \rightarrow o}. \forall X_{\iota}. (C X) \Rightarrow (D X)$$

$$\sqsubseteq^2 = \lambda R_{\iota \rightarrow \iota \rightarrow o}. \lambda S_{\iota \rightarrow \iota \rightarrow o}. \forall X_{\iota}. \forall Y_{\iota}. (R X Y) \Rightarrow (S X Y)$$

$$\text{Dis} = \lambda R_{\iota \rightarrow \iota \rightarrow o}. \lambda S_{\iota \rightarrow \iota \rightarrow o}. \neg \exists X_{\iota}. \exists Y_{\iota}. (R X Y) \wedge (S X Y)$$

$$o = \lambda R_{\iota \rightarrow \iota \rightarrow o}. \lambda S_{\iota \rightarrow \iota \rightarrow o}. \lambda X_{\iota}. \lambda Y_{\iota}. \exists Z_{\iota}. (R X Z) \wedge (S Z Y)$$

- Class expressions become terms of type $\iota \rightarrow o$
- Class membership becomes class application $(C a)$
- Role expressions become terms of type $\iota \rightarrow \iota \rightarrow o$
- Role membership becomes role application $(R a b)$

As we have seen before:

OWL connectives are just abbreviations of HOL terms

[Benzmüller, Research Proposal]

$$\models^{ML} s \quad \text{iff} \quad \models^{HOL} \text{valid } s_{t \rightarrow o}$$

- **Propositional Multimodal Logics** [BenzmüllerPaulson, Log.J.IGPL, 2010]
- **Quantified Multimodal Logics** [BenzmüllerPaulson, CLIMA XI, 2010]
[Benzmüller, AMAI, 2011], [BenzmüllerPaulson, Logica Universal., to appear]
- **Propositional & Quantified Conditional Logics**
[BenzmüllerGenovese, NCMPL, 2011], [BenzmüllerGenoveseGabbayRispoli, submitted, 2011]
- **Temporal Logics:** use semantic modeling of 'irreflexive'
- **Intuitionistic Logics:** [BenzmüllerPaulson, Log.J.IGPL, 2010]
- **Access Control Logics:** [Benzmüller, IFIP SEC, 2009]

Work in progress

- **Spatial Reasoning 'RCC'** [Benzmüller, AMAI, 2011]
- **Semantic Web Language 'OWL'** [Benzmüller, Research Proposal, 2010]
- ...



Automation of Logics and Logic Combinations in HOL

A Simple Example

Problem in Multimodal Logic K

$$\neg(\Box_a \Box_b P) \vee \Box_a P$$

Problem in HOL

valid $\neg(\Box_a \Box_b P) \vee \Box_a P$

A Simple Example

Problem in Multimodal Logic K

Problem in HOL

expanded abbreviation

$\neg(\Box_a \Box_b P) \vee \Box_a P$

valid $\neg(\Box_a \Box_b P) \vee \Box_a P$

$\forall W_t. (\neg(\Box_a \Box_b P) \vee \Box_a P) W$

A Simple Example

Problem in Multimodal Logic K

$$\neg(\Box_a \Box_b P) \vee \Box_a P$$

Problem in HOL

valid $\neg(\Box_a \Box_b P) \vee \Box_a P$

$$\forall W_t. (\neg(\Box_a \Box_b P) \vee \Box_a P) W$$

$$\forall W_t. (\neg(\Box_a \Box_b P) W) \vee (\Box_a P W)$$

expanded abbreviation

A Simple Example

Problem in Multimodal Logic K

$$\neg(\Box_a \Box_b P) \vee \Box_a P$$

Problem in HOL

expanded abbreviation

valid $\neg(\Box_a \Box_b P) \vee \Box_a P$

$$\forall W_t. (\neg(\Box_a \Box_b P) \vee \Box_a P) W$$

$$\forall W_t. (\neg(\Box_a \Box_b P) W) \vee (\Box_a P W)$$

$$\forall W_t. \neg(\Box_a \Box_b P W) \vee (\Box_a P W)$$

A Simple Example

Problem in Multimodal Logic K

$$\neg(\Box_a \Box_b P) \vee \Box_a P$$

Problem in HOL

valid $\neg(\Box_a \Box_b P) \vee \Box_a P$

expanded abbreviation

$$\forall W_t. (\neg(\Box_a \Box_b P) \vee \Box_a P) W$$

$$\forall W_t. (\neg(\Box_a \Box_b P) W) \vee (\Box_a P W)$$

$$\forall W_t. \neg(\Box_a \Box_b P W) \vee (\Box_a P W)$$

$$\forall W_t. \neg(\Box_a \Box_b P W) \vee (\forall V_t. \neg(a W V) \vee (P W))$$

A Simple Example

Problem in Multimodal Logic K

$$\neg(\Box_a \Box_b P) \vee \Box_a P$$

Problem in HOL

valid $\neg(\Box_a \Box_b P) \vee \Box_a P$

expanded abbreviation

$$\forall W_{\iota}. (\neg(\Box_a \Box_b P) \vee \Box_a P) W$$

$$\forall W_{\iota}. (\neg(\Box_a \Box_b P) W) \vee (\Box_a P W)$$

$$\forall W_{\iota}. \neg(\Box_a \Box_b P W) \vee (\Box_a P W)$$

$$\forall W_{\iota}. \neg(\Box_a \Box_b P W) \vee (\forall V_{\iota}. \neg(a W V) \vee (P W))$$

$$\forall W_{\iota}. \neg(\Box_a \Box_b P W) \vee (\forall V_{\iota}. \neg(a_{\iota \rightarrow \iota \rightarrow o} W V) \vee (P_{\iota \rightarrow o} W))$$

A Simple Example

Problem in Multimodal Logic K

$$\neg(\Box_a \Box_b P) \vee \Box_a P$$

Problem in HOL

valid $\neg(\Box_a \Box_b P) \vee \Box_a P$

$$\forall W_t. (\neg(\Box_a \Box_b P) \vee \Box_a P) W$$

$$\forall W_t. (\neg(\Box_a \Box_b P) W) \vee (\Box_a P W)$$

$$\forall W_t. \neg(\Box_a \Box_b P W) \vee (\Box_a P W)$$

$$\forall W_t. \neg(\Box_a \Box_b P W) \vee (\forall V_t. \neg(a W V) \vee (P W))$$

$$\forall W_t. \neg(\Box_a \Box_b P W) \vee (\forall V_t. \neg(a_{t \rightarrow o} W V) \vee (P_{t \rightarrow o} W))$$

...

expanded abbreviation



A Simple Example

Problem in Multimodal Logic K

$$\neg(\Box_a \Box_b P) \vee \Box_a P$$

Problem in HOL

$$\text{valid } \neg(\Box_a \Box_b P) \vee \Box_a P$$

$$\forall W_{l^*}. (\neg(\Box_a \Box_b P) \vee \Box_a P) W$$

$$\forall W_{l^*}. (\neg(\Box_a \Box_b P) W) \vee (\Box_a P W)$$

$$\forall W_{l^*}. \neg(\Box_a \Box_b P W) \vee (\Box_a P W)$$

$$\forall W_{l^*}. \neg(\Box_a \Box_b P W) \vee (\forall V_{l^*}. \neg(a W V) \vee (P W))$$

$$\forall W_{l^*}. \neg(\Box_a \Box_b P W) \vee (\forall V_{l^*}. \neg(a_{l \rightarrow l \rightarrow o} W V) \vee (P_{l \rightarrow o} W))$$

...

$$\forall W_{l^*}. \neg(\dots W) \vee (\forall V_{l^*}. \neg(a_{l \rightarrow l \rightarrow o} W V) \vee (P_{l \rightarrow o} W))$$

A Simple Example

Problem in Multimodal Logic K

$$\neg(\Box_a \Box_b P) \vee \Box_a P$$

Problem in HOL

$$\text{valid } \neg(\Box_a \Box_b P) \vee \Box_a P$$

$$\forall W_{l \bullet}. (\neg(\Box_a \Box_b P) \vee \Box_a P) W$$

$$\forall W_{l \bullet}. (\neg(\Box_a \Box_b P) W) \vee (\Box_a P W)$$

$$\forall W_{l \bullet}. \neg(\Box_a \Box_b P W) \vee (\Box_a P W)$$

$$\forall W_{l \bullet}. \neg(\Box_a \Box_b P W) \vee (\forall V_{l \bullet}. \neg(a W V) \vee (P W))$$

$$\forall W_{l \bullet}. \neg(\Box_a \Box_b P W) \vee (\forall V_{l \bullet}. \neg(a_{l \rightarrow o} W V) \vee (P_{l \rightarrow o} W))$$

...

$$\forall W_{l \bullet}. \neg(\dots\dots\dots W) \vee (\forall V_{l \bullet}. \neg(a_{l \rightarrow o} W V) \vee (P_{l \rightarrow o} W))$$

HOL model finder Nitpick (IsabelleN) quickly finds a countermodel

A Simple Example

Problem in Multimodal Logic K

$$\neg(\Box_a \Box_b P) \vee \Box_a P$$

Problem in HOL

$$\text{valid } \neg(\Box_a \Box_b P) \vee \Box_a P$$

$$\forall W_{l.} (\neg(\Box_a \Box_b P) \vee \Box_a P) W$$

$$\forall W_{l.} (\neg(\Box_a \Box_b P) W) \vee (\Box_a P W)$$

$$\forall W_{l.} \neg(\Box_a \Box_b P W) \vee (\Box_a P W)$$

$$\forall W_{l.} \neg(\Box_a \Box_b P W) \vee (\forall V_{l.} \neg(a W V) \vee (P W))$$

$$\forall W_{l.} \neg(\Box_a \Box_b P W) \vee (\forall V_{l.} \neg(a_{l \rightarrow o} W V) \vee (P_{l \rightarrow o} W))$$

...

$$\forall W_{l.} \neg(\dots\dots\dots W) \vee (\forall V_{l.} \neg(a_{l \rightarrow o} W V) \vee (P_{l \rightarrow o} W))$$

HOL model finder Nitpick (IsabelleN) quickly finds a countermodel

Countermodel for

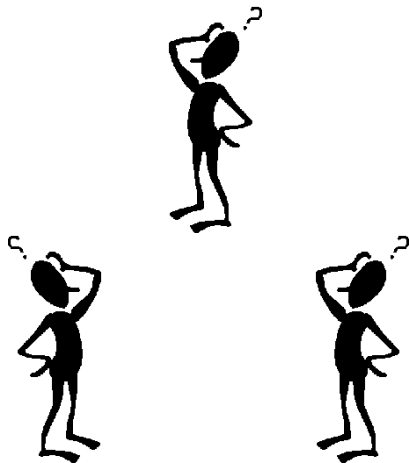
$$\neg(\Box_a \Box_b P) \vee \Box_a P$$

Exemplary study of combinations of logics

- Agent scenarios (e.g. Wise Men Puzzle)
 - common knowledge & knowledge of single agents & time
[Benzmüller, AMAI, 2011]
- Novel combinations
 - knowledge of agents & spatial reasoning
[Benzmüller, DagstuhlSeminar, 2010]
- Combinations that are **relevant for expressive ontologies (SUMO)**
 - knowledge & belief & time & spatial reasoning & ...
[BenzmüllerPease, ARCOE-WS@ECAI, 2010]
[BenzmüllerPease, PAAR-WS@IJCAR, 2010]
[BenzmüllerPease, AICOM, to appear]
[BenzmüllerPease, JWS, in revision]

Wise Men Puzzle

Once upon a time, a king wanted to find the wisest out of his three wisest men. He arranged them in a circle and told them that he would put a white or a black spot on their foreheads and that one of the three spots would certainly be white. The three wise men could see and hear each other but, of course, they could not see their faces reflected anywhere. The king, then, asked to each of them to find out the color of his own spot. After a while, the wisest correctly answered that his spot was white.



Wise Men Puzzle

Once upon a time, a king wanted to find the wisest out of his three wisest men. He arranged them in a circle and told them that he would put a white or a black spot on their foreheads and that one of the three spots would certainly be white. The three wise men could see and hear each other but, of course, they could not see their faces reflected anywhere. The king, then, asked to each of them to find out the color of his own spot. After a while, the wisest correctly answered that his spot was white.

	A	B	C	Answer
1	*	b	b	Aw
2	b	*	b	Bw
3	b	b	*	Cw
4	*	w	b	Aw (→ 2.)
5	*	b	w	Aw (→ 3.)
6	w	*	b	Bw (→ 1.)
7	b	*	w	Bw (→ 3.)
8	w	b	*	Cw (→ 1.)
9	b	w	*	Cw (→ 2.)
10	*	w	w	Aw (→ 3.)
11	w	*	w	Bw (→ 8.)
12	w	w	*	Cw (→ 4.)

Wise Men Puzzle

Once upon a time, a king wanted to find the wisest out of his three wisest men. He arranged them in a circle and told them that he would put a white or a black spot on their foreheads and that one of the three spots would certainly be white. The three wise men could see and hear each other but, of course, they could not see their faces reflected anywhere. The king, then, asked to each of them to find out the color of his own spot. After a while, the wisest correctly answered that his spot was white.

(formalization adapted from: [Balconi, PhD, 1998])

- epistemic modalities (knowledge):
 - \Box_a, \Box_b, \Box_c : individual knowledge of the men
 - \Box_{fool} : common knowledge
- predicate symbol:
 - ws : 'has white spot'
- axioms for common knowledge (S4)

$$\text{valid } \forall^P \phi. \Box_{fool} \phi \Rightarrow \phi \quad (M)$$

$$\text{valid } \forall^P \phi. \Box_{fool} \phi \Rightarrow \Box_{fool} \Box_{fool} \phi \quad (4)$$

inclusion axioms

$$\text{valid } \forall^P \phi. \Box_{fool} \phi \Rightarrow \Box_X \phi$$

$$X \in \{a, b, c\}$$

Wise Men Puzzle

Once upon a time, a king wanted to find the wisest out of his three wisest men. He arranged them in a circle and told them that he would put a white or a black spot on their foreheads and that one of the three spots would certainly be white. The three wise men could see and hear each other but, of course, they could not see their faces reflected anywhere. The king, then, asked to each of them to find out the color of his own spot. After a while, the wisest correctly answered that his spot was white.

(formalization adapted from: [Baltoni, PhD, 1998])

- common knowledge:

at least one of the men has a white spot

$$\text{valid } \Box_{\text{fool}} (ws\ a) \vee (ws\ b) \vee (ws\ c)$$

if X has a white spot, then Y knows this

$$\text{valid } \Box_{\text{fool}} (ws\ X) \Rightarrow \Box_Y (ws\ X)$$

$$X \neq Y \in \{a, b, c\}$$

if X does not have a white spot, then Y knows this

$$\text{valid } \Box_{\text{fool}} \neg (ws\ X) \Rightarrow \Box_Y \neg (ws\ X)$$

$$X \neq Y \in \{a, b, c\}$$

Wise Men Puzzle

Once upon a time, a king wanted to find the wisest out of his three wisest men. He arranged them in a circle and told them that he would put a white or a black spot on their foreheads and that one of the three spots would certainly be white. The three wise men could see and hear each other but, of course, they could not see their faces reflected anywhere. The king, then, asked to each of them to find out the color of his own spot. After a while, the wisest correctly answered that his spot was white.

(formalization adapted from: [Balconi, PhD, 1998])

- if X knows ϕ , then Y knows that X knows ϕ

$$\text{valid } \forall^P \phi. (\Box_X \phi \Rightarrow \Box_Y \Box_X \phi)$$

$$X \neq Y \in \{a, b, c\}$$

- if X does not know ϕ , then Y knows that X does not know ϕ

$$\text{valid } \forall^P \phi. (\neg \Box_X \phi \Rightarrow \Box_Y \neg \Box_X \phi)$$

$$X \neq Y \in \{a, b, c\}$$

Wise Men Puzzle

Once upon a time, a king wanted to find the wisest out of his three wisest men. He arranged them in a circle and told them that he would put a white or a black spot on their foreheads and that one of the three spots would certainly be white. The three wise men could see and hear each other but, of course, they could not see their faces reflected anywhere. The king, then, asked to each of them to find out the color of his own spot. After a while, the wisest correctly answered that his spot was white.

(formalization adapted from: [Balconi, PhD, 1998])

- a does not know his spot is white

$$\text{valid } \neg \Box_a (ws\ a)$$

- b does not know his spot is white

$$\text{valid } \neg \Box_b (ws\ b)$$

- to prove: c does know, that he has a white spot

$$\dots \vdash^{HOL} \text{valid } \Box_c (ws\ c)$$

Wise Men Puzzle

Once upon a time, a king wanted to find the wisest out of his three wisest men. He arranged them in a circle and told them that he would put a white or a black spot on their foreheads and that one of the three spots would certainly be white. The three wise men could see and hear each other but, of course, they could not see their faces reflected anywhere. The king, then, asked to each of them to find out the color of his own spot. After a while, the wisest correctly answered that his spot was white.

(formalization adapted from: [Balconi, PhD, 1998])

- a does not know his spot is white

$$\text{valid } \neg \Box_a (ws\ a)$$

- b does not know his spot is white

$$\text{valid } \neg \Box_b (ws\ b)$$

- to prove: c does know, that he has a white spot

$$\dots \vdash^{HOL} \text{valid } \Box_c (ws\ c)$$

LEO-II can do this effectively

Wise Men Puzzle

Once upon a time, a king wanted to find the wisest out of his three wisest men. He arranged them in a circle and told them that he would put a white or a black spot on their foreheads and that one of the three spots would certainly be white. The three wise men could see and hear each other but, of course, they could not see their faces reflected anywhere. The king, then, asked to each of them to find out the color of his own spot. After a while, the wisest correctly answered that his spot was white.

- temporal modality (time):
 \Box_t : 'in the future it will be the case that'
- after waiting some time, two wise men still don't know the color of their spot

$$\text{valid } \Box_t \neg \Box_a (ws a)$$

$$\text{valid } \Box_t \Box_t \neg \Box_b (ws b)$$

- shortly later the third wise men then knows the color of his spot

$$\dots \vdash^{HOL} \text{valid } \Box_t \Box_t \Box_t \Box_c (ws c)$$

Wise Men Puzzle

Once upon a time, a king wanted to find the wisest out of his three wisest men. He arranged them in a circle and told them that he would put a white or a black spot on their foreheads and that one of the three spots would certainly be white. The three wise men could see and hear each other but, of course, they could not see their faces reflected anywhere. The king, then, asked to each of them to find out the color of his own spot. After a while, the wisest correctly answered that his spot was white.

- wait a second: \Box_t has not yet been characterized as temporal modality

relation t is transitive: $\forall^P \phi. \Box_t \phi \Rightarrow \Box_t \Box_t \phi$

relation t is irreflexive: $\boxed{???$

Wise Men Puzzle

Once upon a time, a king wanted to find the wisest out of his three wisest men. He arranged them in a circle and told them that he would put a white or a black spot on their foreheads and that one of the three spots would certainly be white. The three wise men could see and hear each other but, of course, they could not see their faces reflected anywhere. The king, then, asked to each of them to find out the color of his own spot. After a while, the wisest correctly answered that his spot was white.

- wait a second: \Box_t has not yet been characterized as temporal modality

relation t is transitive: $\forall^P \phi. \Box_t \phi \Rightarrow \Box_t \Box_t \phi$

relation t is irreflexive: (irreflexive t)

$$\text{irreflexive} = \lambda R_{\iota \rightarrow \iota} \sigma. \forall W_{\iota}. \neg (R W W)$$

Wise Men Puzzle

Once upon a time, a king wanted to find the wisest out of his three wisest men. He arranged them in a circle and told them that he would put a white or a black spot on their foreheads and that one of the three spots would certainly be white. The three wise men could see and hear each other but, of course, they could not see their faces reflected anywhere. The king, then, asked to each of them to find out the color of his own spot. After a while, the wisest correctly answered that his spot was white.

- wait a second: \Box_t has not yet been characterized as temporal modality

relation t is transitive: (transitive t)

relation t is irreflexive: (irreflexive t)

$$\text{irreflexive} = \lambda R_{\iota \rightarrow \iota \rightarrow o}. \forall W_{\iota}. \neg (R W W)$$

Wise Men Puzzle

Once upon a time, a king wanted to find the wisest out of his three wisest men. He arranged them in a circle and told them that he would put a white or a black spot on their foreheads and that one of the three spots would certainly be white. The three wise men could see and hear each other but, of course, they could not see their faces reflected anywhere. The king, then, asked to each of them to find out the color of his own spot. After a while, the wisest correctly answered that his spot was white.

- wait a second: \Box_t has not yet been characterized as temporal modality

relation t is transitive: (transitive t)

relation t is irreflexive: (irreflexive t)

$$\text{irreflexive} = \lambda R_{\iota \rightarrow \iota \rightarrow o}. \forall W_{\iota}. \neg (R W W)$$

LEO-II can solve the modified puzzle effectively

Region Connection Calculus (RCC)
as fragment of HOL:

[RandellCuiCohn, 1992]

disconnected : $dc = \lambda X_{\tau}. \lambda Y_{\tau}. \neg(c X Y)$

part of : $p = \lambda X_{\tau}. \lambda Y_{\tau}. \forall Z. ((c Z X) \Rightarrow (c Z Y))$

identical with : $eq = \lambda X_{\tau}. \lambda Y_{\tau}. ((p X Y) \wedge (p Y X))$

overlaps : $o = \lambda X_{\tau}. \lambda Y_{\tau}. \exists Z. ((p Z X) \wedge (p Z Y))$

partially o : $po = \lambda X_{\tau}. \lambda Y_{\tau}. ((o X Y) \wedge \neg(p X Y) \wedge \neg(p Y X))$

ext. connected : $ec = \lambda X_{\tau}. \lambda Y_{\tau}. ((c X Y) \wedge \neg(o X Y))$

proper part : $pp = \lambda X_{\tau}. \lambda Y_{\tau}. ((p X Y) \wedge \neg(p Y X))$

tangential pp : $tpp = \lambda X_{\tau}. \lambda Y_{\tau}. ((pp X Y) \wedge \exists Z. ((ec Z X) \wedge (ec Z Y)))$

nontang. pp : $ntpp = \lambda X_{\tau}. \lambda Y_{\tau}. ((pp X Y) \wedge \neg \exists Z. ((ec Z X) \wedge (ec Z Y)))$

A trivial problem for RCC:

Catalunya is a border region of Spain	$(tpp \text{ catalunya } \text{spain}),$
Spain and France share a border	$(ec \text{ spain } \text{france}),$
Paris is a region inside France	$(ntpp \text{ paris } \text{france})$

\vdash^{HOL}

Catalunya and Paris are disconnected	$(dc \text{ catalunya } \text{paris})$
	\wedge
Spain and Paris are disconnected	$(dc \text{ spain } \text{paris})$

[Benzmüller, AMAI, 2011]

A trivial problem for RCC:

Catalunya is a border region of Spain	$(tpp \text{ catalunya spain}),$
Spain and France share a border	$(ec \text{ spain france}),$
Paris is a region inside France	$(ntpp \text{ paris france})$

$$\vdash_{2.3s}^{\text{HOL}}$$

Catalunya and Paris are disconnected	$(dc \text{ catalunya paris})$
	\wedge
Spain and Paris are disconnected	$(dc \text{ spain paris})$

[Benzmüller, AMAI, 2011]

valid $\forall \phi. \Box_{\text{fool}} \phi \supset \Box_{\text{bob}} \phi,$

valid $\Box_{\text{fool}} (\lambda W. (ec \text{ spain } france)),$

valid $\Box_{\text{bob}} (\lambda W. (tpp \text{ catalunya } spain)),$

valid $\Box_{\text{bob}} (\lambda W. (ntpp \text{ paris } france))$

\vdash^{HOL} valid $\Box_{\text{bob}} (\lambda W. ((dc \text{ catalunya } paris) \wedge (dc \text{ spain } paris)))$

valid $\forall \phi. \Box_{\text{fool}} \phi \supset \Box_{\text{bob}} \phi,$
valid $\Box_{\text{fool}} (\lambda W. (ec \text{ spain } france)),$
valid $\Box_{\text{bob}} (\lambda W. (tpp \text{ catalunya } spain)),$
valid $\Box_{\text{bob}} (\lambda W. (ntpp \text{ paris } france))$
 $\vdash_{20.4s}^{HOL}$ valid $\Box_{\text{bob}} (\lambda W. ((dc \text{ catalunya } paris) \wedge (dc \text{ spain } paris)))$

- valid $\forall \phi. \Box_{\text{fool}} \phi \supset \Box_{\text{bob}} \phi,$
 valid $\Box_{\text{fool}} (\lambda W. (ec \text{ spain } france)),$
 valid $\Box_{\text{bob}} (\lambda W. (tpp \text{ catalunya } spain)),$
 valid $\Box_{\text{bob}} (\lambda W. (ntpp \text{ paris } france))$
 $\vdash_{20.4s}^{HOL}$ valid $\Box_{\text{bob}} (\lambda W. ((dc \text{ catalunya } paris) \wedge (dc \text{ spain } paris)))$
 $\not\vdash^{HOL}$ valid $\Box_{\text{fool}} (\lambda W. ((dc \text{ catalunya } paris) \wedge (dc \text{ spain } paris)))$

- valid $\forall \phi. \Box_{\text{fool}} \phi \supset \Box_{\text{bob}} \phi,$
 valid $\Box_{\text{fool}} (\lambda W. (ec \text{ spain } france)),$
 valid $\Box_{\text{bob}} (\lambda W. (tpp \text{ catalunya } spain)),$
 valid $\Box_{\text{bob}} (\lambda W. (ntpp \text{ paris } france))$
 $\vdash_{20.4s}^{HOL}$ valid $\Box_{\text{bob}} (\lambda W. ((dc \text{ catalunya } paris) \wedge (dc \text{ spain } paris)))$
 $\not\vdash_{39.7s}^{HOL}$ valid $\Box_{\text{fool}} (\lambda W. ((dc \text{ catalunya } paris) \wedge (dc \text{ spain } paris)))$

valid $\forall \phi. \Box_{\text{fool}} \phi \supset \Box_{\text{bob}} \phi,$
 valid $\Box_{\text{fool}} (\lambda W. (ec \text{ spain } france)),$
 valid $\Box_{\text{bob}} (\lambda W. (tpp \text{ catalunya } spain)),$
 valid $\Box_{\text{bob}} (\lambda W. (ntpp \text{ paris } france))$
 $\vdash_{20.4s}^{HOL}$ valid $\Box_{\text{bob}} (\lambda W. ((dc \text{ catalunya } paris) \wedge (dc \text{ spain } paris)))$
 $\not\vdash_{39.7s}^{HOL}$ valid $\Box_{\text{fool}} (\lambda W. ((dc \text{ catalunya } paris) \wedge (dc \text{ spain } paris)))$

Key idea is “Lifting” of RCC propositions to modal predicates:

$$\underbrace{(tpp \text{ catalunya } spain)}_{\text{type } o} \longrightarrow \underbrace{(\lambda W. (tpp \text{ catalunya } spain))}_{\text{type } \iota \rightarrow o}$$



Automation of Meta-Properties of Logics in HOL

Correspondences between properties of accessibility relations like

$$\text{symmetric} = \lambda R. \forall S, T. R S T \Rightarrow R T S$$

$$\text{serial} = \lambda R. \forall S. \exists T. R S T$$

and corresponding axioms

$$\begin{aligned} \forall R. \text{symmetric } R &\Leftrightarrow \\ &\Rightarrow \text{valid } \forall^P \phi. \phi \supset \Box_R \Diamond_R \phi \quad (B) \end{aligned}$$

$$\begin{aligned} \forall R. \text{serial } R &\Leftrightarrow \\ &\Rightarrow \text{valid } \forall^P \phi. \Box_R \phi \supset \Diamond_R \phi \quad (D) \end{aligned}$$

Correspondences between properties of accessibility relations like

$$\text{symmetric} = \lambda R. \forall S, T. R S T \Rightarrow R T S$$

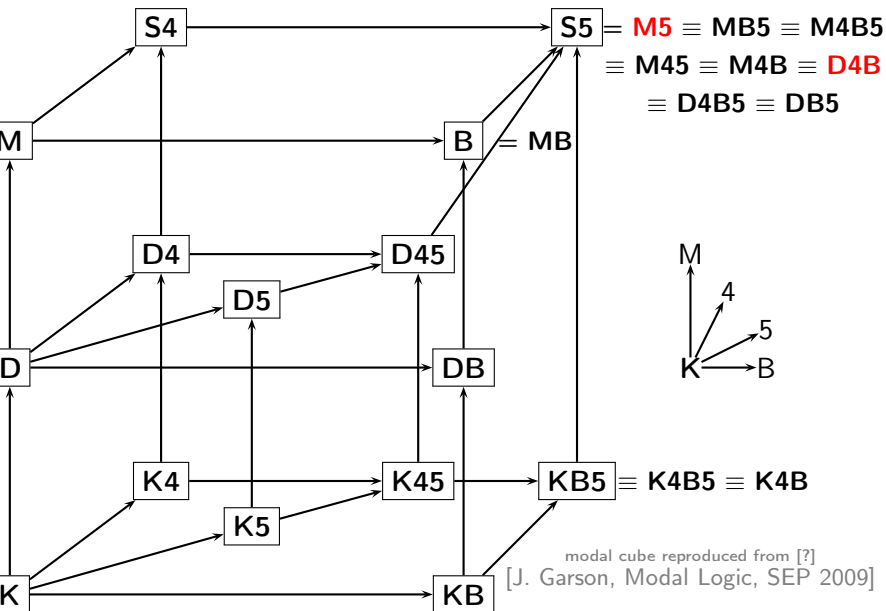
$$\text{serial} = \lambda R. \forall S. \exists T. R S T$$

and corresponding axioms

$$\forall R. \text{symmetric } R \stackrel{0,0s}{\Leftarrow} \stackrel{0,0s}{\Rightarrow} \text{valid } \forall^P \phi. \phi \supset \square_R \diamond_R \phi \quad (B)$$

$$\forall R. \text{serial } R \stackrel{0,0s}{\Leftarrow} \stackrel{0,0s}{\Rightarrow} \text{valid } \forall^P \phi. \square_R \phi \supset \diamond_R \phi \quad (D)$$

Such proofs — including axioms D, M, 4, B, 5 — can be automated with LEO-II in no-time!



$\forall R.$

$$\wedge \left. \begin{array}{l} \text{valid } \forall^P \phi. \Box_R \phi \supset \phi \\ \text{valid } \forall^P \phi. \Diamond_R \phi \supset \Box_R \Diamond_R \phi \end{array} \right\} M5$$

 \Leftrightarrow

$$\wedge \left. \begin{array}{l} \text{valid } \forall^P \phi. \Box_R \phi \supset \Diamond_R \phi \\ \text{valid } \forall^P \phi. \Box_R \phi \supset \Box_R \Box_R \phi \\ \text{valid } \forall^P \phi. \phi \supset \Box_R \Diamond_R \phi \end{array} \right\} D4B$$

[Benzmüller, Festschrift Walther, 2010]

$\forall R.$

$$\wedge \left. \begin{array}{l} \text{valid } \forall^P \phi. \Box_R \phi \supset \phi \\ \text{valid } \forall^P \phi. \Diamond_R \phi \supset \Box_R \Diamond_R \phi \end{array} \right\} M5$$

 \Leftrightarrow

$$\wedge \left. \begin{array}{l} \text{serial } R \\ \text{valid } \forall^P \phi. \Box_R \phi \supset \Box_R \Box_R \phi \\ \text{symmetric } R \end{array} \right\} D4B$$

$\forall R.$ \wedge reflexive R
 \wedge euclidean R } M5 \Leftrightarrow \wedge serial R
 \wedge transitive R
 \wedge symmetric R } D4B

$\forall R.$

\wedge reflexive R
 \wedge euclidean R } M5

0.1s
 \Leftrightarrow

serial R
 \wedge transitive R
 \wedge symmetric R } D4B

Proof with LEO-II in 0.1s

[Benzmüller, Festschrift Walther, 2010]

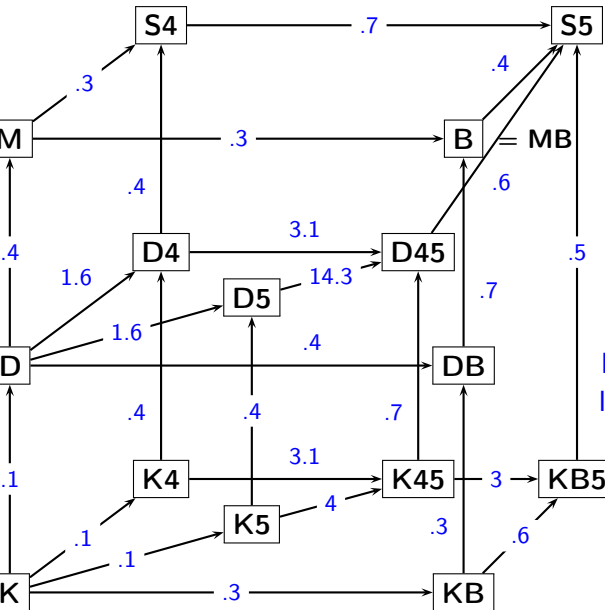
$\forall R.$

\wedge reflexive R
 \wedge euclidean R } M5

 \Leftrightarrow

\wedge serial R
 \wedge transitive R
 \wedge symmetric R } D4B

[Benzmüller, Festschrift Walther, 2010]



$$\begin{aligned}
 &= M5 \stackrel{.1}{\equiv} MB5 \stackrel{.2}{\equiv} M4B5 \\
 &\stackrel{.2}{\equiv} M45 \stackrel{.1}{\equiv} M4B \stackrel{.1}{\equiv} D4B \\
 &\stackrel{.2}{\equiv} D4B5 \stackrel{.1}{\equiv} DB5
 \end{aligned}$$

fastest results by:

- LEO-II (prover)
- Satallax (prover)
- TPS (prover)
- Satallax (mod.find.)
- IsabelleN (mod.find.)
- IsabelleM (mod.find.)

$$\stackrel{.2}{\equiv} K4B5 \stackrel{.1}{\equiv} K4B$$

$\Sigma < 40\text{sec.}$

Automation of meta-level properties

[Benzmüller, Festschrift Walther, 2010]

- Correspondences between axioms and semantic properties
 $\text{valid } \forall \phi. \Box_r \phi \supset \Box_r \Box_r \phi$ (axiom 4)
 \Leftrightarrow (transitive r)
- Dependence/independence of axioms
 base modal logic $K \not\models$ axiom 4?
- Consistency of logics and logic combinations
 Is logic $S4$ ($K+M+4$) consistent?
- Inclusion/non-inclusion relations between logics
 Is logic $K45$ ($K+M+5$) included in logic $S4$ ($K+M+4$)? Why not?

Running experiments (thousands of problems): exploration of

- Modal Logics with Geoff Sutcliffe
- Conditional Logics with Valerio Genovese, Dov Gabbay

[Segerberg, 1973]

- [Segerberg, 1973] discusses a 2-dimensional logic providing two S5 modalities \Box_a and \Box_b .
- He adds further axioms stating that these modalities are commutative and orthogonal.
- It actually turns out that orthogonality is already implied in this context.

reflexive a , transitive a , euclid. a ,

reflexive b , transitive b , euclid. b ,

valid $\forall \phi. \Box_a \Box_b \phi \Leftrightarrow \Box_b \Box_a \phi$

\vdash^{HOL}

valid $\forall \phi, \psi. \Box_a (\Box_a \phi \vee \Box_b \psi) \supset (\Box_a \phi \vee \Box_a \psi)$

\wedge valid $\forall \phi, \psi. \Box_b (\Box_a \phi \vee \Box_b \psi) \supset (\Box_b \phi \vee \Box_b \psi)$

[Benzmüller, AMAI, 2011]

[Segerberg, 1973]

- [Segerberg, 1973] discusses a 2-dimensional logic providing two S5 modalities \Box_a and \Box_b .
- He adds further axioms stating that these modalities are commutative and orthogonal.
- It actually turns out that orthogonality is already implied in this context.

reflexive a , transitive a , euclid. a ,

reflexive b , transitive b , euclid. b ,

valid $\forall \phi. \Box_a \Box_b \phi \Leftrightarrow \Box_b \Box_a \phi$

\vdash^{HOL}

valid $\forall \phi, \psi. \Box_a (\Box_a \phi \vee \Box_b \psi) \supset (\Box_a \phi \vee \Box_a \psi)$

\wedge valid $\forall \phi, \psi. \Box_b (\Box_a \phi \vee \Box_b \psi) \supset (\Box_b \phi \vee \Box_b \psi)$

[Benzmüller, AMAI, 2011]

[Segerberg, 1973]

- [Segerberg, 1973] discusses a 2-dimensional logic providing two S5 modalities \Box_a and \Box_b .
- He adds further axioms stating that these modalities are commutative and orthogonal.
- It actually turns out that orthogonality is already implied in this context.

reflexive a , transitive a , euclid. a ,

reflexive b , transitive b , euclid. b ,

valid $\forall \phi. \Box_a \Box_b \phi \Leftrightarrow \Box_b \Box_a \phi$

\vdash^{HOL}

valid $\forall \phi, \psi. \Box_a (\Box_a \phi \vee \Box_b \psi) \supset (\Box_a \phi \vee \Box_a \psi)$

\wedge valid $\forall \phi, \psi. \Box_b (\Box_a \phi \vee \Box_b \psi) \supset (\Box_b \phi \vee \Box_b \psi)$

[Benzmüller, AMAI, 2011]

[Segerberg, 1973]

- [Segerberg, 1973] discusses a 2-dimensional logic providing two S5 modalities \Box_a and \Box_b .
- He adds further axioms stating that these modalities are commutative and orthogonal.
- It actually turns out that orthogonality is already implied in this context.

reflexive a , transitive a , euclid. a ,

reflexive b , transitive b , euclid. b ,

valid $\forall \phi. \Box_a \Box_b \phi \Leftrightarrow \Box_b \Box_a \phi$

\vdash^{HOL}

proof by LEO-II in 0.2s

valid $\forall \phi, \psi. \Box_a (\Box_a \phi \vee \Box_b \psi) \supset (\Box_a \phi \vee \Box_a \psi)$

\wedge valid $\forall \phi, \psi. \Box_b (\Box_a \phi \vee \Box_b \psi) \supset (\Box_b \phi \vee \Box_b \psi)$

[Benzmüller, AMAI, 2011]

Essential: TPTP infrastructure for HOL (tptp.org) (with G. Sutcliffe)

- project result of: EU FP7 IIF grant THFTPTP
- standardized THF syntax for HOL (& more)
- problem library
- prover competition
- online access ≥ 6 ATPs/model finders
- tools for proof verification, ...

www.tptp.org \rightarrow PUZ087^{1/2}.p (Wise Men Puzzle)

[SutcliffeBenzmüller, Journal of Formalized Reasoning, 2010]

Core Questions:

- 1 Classical Higher Order Logic (HOL) as Universal Logic?
 - 2 HOL Provers & Model Finders as Generic Reasoning Tools?
 - 3 Combinations with Specialist Reasoners (if available)?
- (1)&(2) are interesting and relevant: Evidence given in talk!?
 - (3) not further discussed: – ongoing and future work
- My vision is an automated (& interactive) generic logic engine with HOL theorem provers and model finders as backbone, and with integrated, more effective specialist reasoners (if available) as collaborating agents.
- Don't forget: There are many reasons for the automation of HOL!